AD-A062 315 NORTH CAROLINA AGRICULTURAL AND TECHNICAL STATE UNIV --ETC F/G 9/5
A STUDY OF TWO-DIMENSIONAL RECURSIVE DIGITAL FILTERS AND A STUD--ETC(U)
NOV 78 N00014-77-C-0199

UNCLASSIFIED NL

1 OF 2
ADA
062315

NORTH CAROLINA A & T STATE UNIVERSITY

FINAL REPORT
NAVAL AIR SYSTEMS COMMAND
Washington, D. C.

A STUDY OF TWO-DIMENSIONAL RECURSIVE DIGITAL FILTERS AND A STUDY OF IMPROVED STATISTICAL AND NON-DESTRUCTIVE TESTING METHODS OF FAILURE PREDICTION IN BRITTLE MATERIALS

Contract No. N00014-77-C-0199

November 10, 1978

LEVEL II

NAVAIR 3022E

1

New

NORTH CAROLINA A & T STATE UNIV~~ERSITY~~, Greensboro.

9 FINAL REPORT,

NAVAL AIR SYSTEMS COMMAND

Washington, D. C.

6

A STUDY OF TWO-DIMENSIONAL RECURSIVE DIGITAL FILTERS AND A STUDY OF IMPROVED STATISTICAL AND NON-DESTRUCTIVE TESTING METHODS OF FAILURE PREDICTION IN BRITTLE MATERIALS,

15 Contract No. N00014-77-C-0199 new

11 1Ø Nov[redacted]78

12 122p.

This report contains separate topical reports as indicated below:

Project Director: William J. Craft
School of Engineering
N. C. A & T State University
Greensboro, N. C. 27411
(919) 379-7549

SUMMARY

Research efforts covered by the subject contract have been directed toward the stability and synthesis problems for two dimensional digital recursive digital filters. Such filters are advantageous when a premium is placed upon computer storage or computation time.

Previously defined results on stability analysis have been refined and a revised paper has been submitted to the IEEE Transactions on Circuits and Systems for publication. This research effort will continue to address this complex problem.

Previously defined results on synthesis of bandpass and band enhancement filters have been refined. Results of these improvements will be installed in the Naval Intelligence Support Center's Spatial Domain Filtering Package which was designed and implemented by this author. This work will be done under a separate effort. This package has provided very satisfactory results to date and the refinements should significantly improve the overall performance of the package. Documentation on this spatial domain filtering package should be available within the next year.

Approximately circularly symmetric lowpass, highpass, bandpass, band-stop, low frequency boost and high frequency boost filters have been designed. Evaluation of these filters with actual images of various types need to be addressed and has been proposed as a follow on to the effort described in this report.

INTRODUCTION

The two-dimensional recursive digital filter is particularly suited to image processing applications when there is a premium placed on computer memory requirements and time for processing. Due to these considerations, it has definite advantages over the Fast Fourier Transform (FFT) algorithm for many image processing operations[1]. The application of recursive digital filters to image processing, however, has been hampered by two problems: stability and synthesis[2]. The synthesis problem is the problem of expressing the desired impulse response in closed form and thus determing the filtering coefficients. The stability problem occurs because the recursive filter requires feedback of past output values and therefore it can become unstable.

The research for the first year has been on both problems with progress made in both areas. This report discusses the progress made in both areas and the directions for future research.

STABILITY

The stability problem for one dimensional digital recursive filters is straight forward. The roots of the polynomial in the closed form of the one dimensional Z-Transform for the filter impulse response must have magnitudes less than one. Stability analysis is therefore reduced to finding roots of nth degree polynomials with constant coefficients. For the two dimensional problem, stability is not straightforward because a two variable polynomial is not generally factorable into distinct roots. When the polynomial in the denominator of the two dimensional Z-Transform[3] for the impulse response is factorable, then the stability analysis procedure is the same as for the one dimensional problem.

The two dimensional stability problem is very complicated if the poly-

nomial in the denominator of the Z-Transform is not factorable into distinct roots. Efforts by other researchers have been directed toward examining regions of roots for two variable polynomials which is computationally feasible only for very simple filters[4].

The method used by this researcher is to express the two dimensional digital recursive equation as a matrix recursive equation. The description of the matrix recursive equation and its derivation is given in Appendix C. The resulting matrix recursive equation has three coefficient matrices, B_1, B_2 and A. Appendix A gives a summary of stability analysis results to date[5]. A paper entitled "Stability Analysis of two dimensional Recursive Filters" by W. E. Alexander and S. A. Pruess was revised as a part of this research effort and resubmitted for publication to the IEEE Transactions on Circuits and Systems. A preprint of this paper is given in Appendix D.

In practice, the stability analysis procedure which only involves finding the spectral radius of a matrix with real coefficients is very simple and easily implemented. Computer algorithms are readily available to perform the necessary computations. The procedure is regularly used by this researcher for stability analysis of two dimensional recursive digital filters.

SYNTHESIS

Often it is possible to express a desired two dimensional digital recursive filter as the product or sum of two one dimensional digital filters. That is the two dimensional Z-Transform of the digital recursive filter can be expressed as the product or sum of two one-dimensional Z-Transforms. In either case, the two dimensional synthesis problem is reduced to the synthesis of two one-dimensional filters. However, it is not possible to design sum

separable or product separable digital recursive filters for all applications. For those applications, the design of the required two dimensional digital recursive filter is considerably more complicated.

Many imaging systems have a natural circular symmetry. In general, the optical transfer function of a circularly symmetric imaging system is uniform with respect to direction. The natural consequence is that filters with circularly symmetric impulse response functions are generally very desirable for image processing. The relationship between circular symmetry of the impulse response and the frequency response dictates that the design requirement is for these filters to have a circularly symmetric frequency response[6].

LOW PASS FILTER DESIGN

The design goal is to design a low pass filter with circularly symmetric frequency magnitude characteristics. No attempt is made to control the phase response of the desired filter. This presents no difficulties in implementing the designed filters because the two pass, linear phase recursive digital filtering procedure can be used to obtain linear phase[5].

The magnitude characteristic for the one dimensional Butterworth approximation filter in the Laplace Transform variable is given by

$$h(s)h(-s) = \frac{1}{1+(-1)^n \varepsilon^2 \left(\frac{s}{\omega_o}\right)^{2n}} \tag{1}$$

The corresponding equation for two dimensional filters is given by

$$h(s_1,s_2) = \frac{1}{1+(-1)^n \varepsilon^2 \left(\frac{s_1^2 + s_2^2}{\omega_x^2 + \omega_y^2}\right)^{n}} \tag{2}$$

where s_1 and ω_x are respectively Laplace Transform and cutoff frequency variables for the x direction and s_2 and ω_y are respectively Laplace Transform and cutoff frequency variables for the y direction.

If the bilinear transformation[7] is applied to (2) to obtain a two dimensional Z-Transform, we obtain

$$H(z,w)^2 = \frac{[(z+1)^2\ (w+1)^2]^n}{[(z+1)^2(w+1)^2]^n + \varepsilon^2(-1)^n C^n[(z-1)^2(w+1)^2+(z+1)^2(w-1)^2]} \tag{3}$$

Let $\quad C = 1/[\tan^2(\omega_R T/2)] \qquad \omega_R{}^T = \omega_x{}^2 T_x{}^2 + \omega_y{}^2 T_y{}^2 \quad (4)$

Note that ω_y is the effective radial cutoff frequency. In continuing the design procedure in manner similar to that used for one dimensional digital recursive filters[8] difficulties are encountered because the denominator of (3) is not factorable in distinct roots of z and w. However, a suitable approximation may be obtained by factoring along the w-z plane. Thus in this plane, one obtains

$$\left|H(z,z)\right|^2 = \frac{(z+1)^{4n}}{(z+1)^{4n}+(-1)^n C^n[2(z-1)^2(z+1)^2]^n} \tag{5}$$

Simplifying, we obtain

$$\left|H(z,z)\right|^2 = \frac{(z+1)^{2n}}{(z+1)^{2n}+\varepsilon^2(-1)^n C^n[2(z-1)^2]^n} \tag{6}$$

Thus the poles of the magnitude response in the w=z plane occur in reciprocal pairs as roots of the denominator of (6).

As with one dimensional filters, consideration should be given to round off errors and truncation errors in implementing two dimensional digital recursive filters. Thus a cascade realization is very desirable

because it acts to minimize round off error. Also, it is desirable to avoid using complex arithmetic when implementing two dimensional recursive filters. This leads to a natural selection of implementing a basic filter with either one pole and one zero or two poles and two zeros to accomodate complex conjugate pairs of poles. Then any general filter would be implemented as stages of the one pole or two pole filter.

If we let n=1 in (6), we obtain a factorization of $\left|H(z,z)\right|^2 = H_x(z)H_x(z^{-1})$ such that for a stable filter design.

$$H_x(z)=\frac{A\ (z+1)}{(z+P)} \tag{7}$$

$$A=\frac{1}{1-2C\varepsilon^2} \tag{8}$$

$$P = \frac{+(1+2C\varepsilon^2\ ) - 2\sqrt{2C\ \varepsilon^2}}{1-2C\varepsilon^2} \tag{9}$$

except that P=0 for $C=0.5/\varepsilon^2$.

For the case where n is equal to or greater than 2, factorization becomes more complicated and the computer is used to find the roots with magnitudes less than one.

Forming the two dimensional Z-Transform for the final low pass filter design for n equal to one, we obtain

$$H_L(z,w) = \frac{A^2\,(z+1)\,(w+1)}{(z+P)\,(w+P)} \tag{10}$$

Note that this filter design is product separable and inherently stable because we have selected P such that $|P|$ is always less than one. In a similar fashion, we can design filters for n greater than one.

BAND PASS FILTER DESIGN

Once a low pass filter has been designed, it is possible to obtain highpass, band pass and band stop filters as well as low frequency boost

and high frequency boost filters from the low pass design. In this section, we discuss the design of a general boost filter which can be used with proper parameter values to obtain the above mentioned filters. With the low pass filter design (n=1) given in (10), we can obtain a filter with the desired magnitude as given by

$$\left|H(z,w)\right| = \alpha + \beta\left|H_L(z,w)\right|^2 \tag{11}$$

where $H_L(z,w) = H_L(z,w)\ H_L(z^{-1},w^{-1})$. Thus

$$H(z,w) = \alpha + \frac{\beta A^4 (z+1)(z^{-1}+1)(w+1)(w\ +1)}{(z+P)(z^{-1}+P)\ (w+P)(w^{-1}+P)} \tag{12}$$

$$\left|H(z,w)\right| = \frac{\alpha[(z+p)(1+Pz)(w+P)(1+Pw)] + \beta A^4\ [(z+1)^2(w+1)^2]}{(z+P)(1+Pz)(w+P)(1+Pw)} \tag{13}$$

Note that the filter represented by (13) is unstable because of the terms (1+Pz) and (1+Pw) in the denominator. The poles corresponding to these terms have magnitudes greater than one since P has a magnitude less than one. However, we can stabilize (13) by using the minimum phase version of the denominator. This does not change the magnitude since the magnitude of (1+Pz) is equal to the magnitude of (z+P) and the magnitude of (1+Pw) is equal to the magnitude of (w+P). Thus the desired boost filter design has the two dimensional Z-Transform

$$H_B(z,w) = \frac{\alpha[Pz^2+(1+P^2)z+P][Pw^2+(1+P^2)w+P] + \beta A^4[(z+1)^2(w+1)^2]}{(z+P)^2(w+P)^2} \tag{14}$$

thus if express (14) in the form

$$H_B(Z,W) = \frac{\sum_{J=0}^{L}\sum_{K=0}^{L} a_{JK}\ z^{-J}w^{-K}}{\sum_{J=0}^{L}\sum_{K=0}^{L} b_{JK}\ z^{-J}w^{-K}} \tag{15}$$

it follows that

$$\begin{aligned} a_{00} &= a_{22} = \alpha P^2 + \beta A^4 \\ a_{10} &= a_{01} = a_{12} = \alpha P(1+P^2) + 2\beta A^4 \\ a_{02} &= a_{20} = \alpha P^2 + \beta A^4 \\ a_{11} &= \alpha(1+P^2)^2 + 4\beta A^4 \end{aligned} \tag{16}$$

$$
\begin{aligned}
b_{00} &= 1.0 \\
b_{01} &= b_{10} = 2P \\
b_{02} &= b_{20} = P^2 \\
b_{12} &= b_{21} = 2P^3 \\
b_{11} &= 4\ P^2 \\
b_{22} &= P^4
\end{aligned} \tag{17}
$$

It only remains to determine the value of ε for both the low pass and boost filters. Note that the squared magnitude of $H_L(Z,W)$ is equal to $1/(1+\varepsilon^2)$ when the radial frequency, ω, is equal to the radial cutoff frequency, ω_R. Thus for n=1 $|H_L(Z,W)|^2 = 1/(1+\varepsilon^2)$ at the cutoff frequency. If we use the double pass linear phase filter which is desirable[5], the magnitude of the resultant filter is the squared magnitude of the orginial filter. If we desire the magnitude of the resulting filter to be down 3db at the cutoff, we obtain

$$\frac{1}{\sqrt{2}} = \frac{1}{(1+\varepsilon 2)^2} \tag{18}$$

The same value for ε^2 is appropriate for the low frequency boost filter.

For the high pass filter, we have $\alpha=1$ and $\beta=-1$. Thus the magnitude of the frequency response for the double pass linear phase filter at the cutoff frequency is given by the relationship

$$\frac{1}{2} = 1 - \frac{1}{(1+\varepsilon^2)}^2 \tag{19}$$

$$\varepsilon^2 = 1.0/(2^{\frac{1}{4}}-1)$$

The same value for ε^2 is appropriate for the high frequency boost filter.

If we designated B as the magnitude of the desired boost, then the

low frequency boost filter values for α and β are given by

$$\alpha = 1.0 ; \qquad \beta = B-1.0 \tag{20}$$

Correspondingly, for the high frequency boost filter, the values of α and β are given by

$$\alpha = B ; \qquad \beta = -B+1.0 \tag{21}$$

Examples of two dimensional recursive filter designs are given in Appendix B.

ROTATED ONE DIMENSIONAL FILTERS

A problem of interest in image processing is to filter with a one dimensional filter with the orientation of the filter specified and independent of the sampling directions. This type of filter would be useful for enhancing or suppressing linear features, for system noise suppression or for image correction (i.e. linear smear). However, any one dimensional digital recursive filter which is rotated becomes a two dimensional filter associated problems in stability and synthesis.

Constraints with regard to angle of rotation and stability of rotated filters have been developed by Costa and Ventsonopoulos[9]. They have used several rotated low pass filters to obtain circularly symmetric lowpass filters. This approach is currently being evaluated with regard to use with single rotated filters.

RECOMMENDATIONS FOR FUTURE RESEARCH

Approximately circularly symmetric band pass and band enhancement filters have been designed. These filter designs must be evaluated with regard to performance on actual images of various types and with regard

to circular symmetry as a function of critical frequency. Methods of improving response and eliminating errors in circular symmetry need to be investigated.

Techniques for obtaining rotated one dimensional digital filters need to be investigated with regard to practical use. The most practical method will be developed into an algorithm for image processing.

The use of recursive digital filters for image correction has been hampered by problems in designing filters with arbitrarily specified magnitude and phase characteristics. Long term future research efforts need to be directed toward this problem.

Recursive digital filters are practical for image processing applications using small computers, minicomputers and special signal processors. Applications range from real time image processing and data acquisition to medical applications and industrial process monitoring. However, special algorithms must be developed for many of these applications. Such practical research problems provide the real payoff for research on two dimensional digital recursive filters.

REFERENCES

[1] Ernest L. Hall, "A Comparison of Computations for Spatial Frequency Filtering", Proceedings of the IEEE, Vol. 60, No. 7, 1972, pp 887-891

[2] N. K. Bose, "Problems and Progress in Multidimensional Systems Theory", Proceedings of the IEEE, Vol. 65, No. 6, 1977, pp. 824-840.

[3] Lawrence R. Rabiner and Bernard Gold, Theory and Applications of Digital Signal Processing, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1975, pp. 442-455.

[4] Dennis Goodman, "Some Stability Properties of the Two Dimensional Linear Shift-Invariant Digital FIlters", IEEE Transactions on Circuits and Systems, Vol. CAS24, No. 4, 1977, pp. 201-208.

[5] Winser E. Alexander, Stability and Synthesis of Two-Dimensional Digital Recursive Filters, Ph.D. Dissertation, University of New Mexico, Albuquerque, N. M., 1974 (University Microfilms, Ann Arbor, Mich.).

[6] A. Papoulis, Systems and Transforms with Applications in Optics, McGraw-Hill, Inc., New York, N. Y., 1968, p. 140.

[7] Bernard Gold and Charles Rader, Digital Processing of Signals, McGraw-Hill, Inc., New York, N. Y., 1969, Chapter 6.

[8] Samuel Stearns, Digital Signal Analysis, Hayden Publishing Company, Rochelle Park, N. J., 1975, Chapter 12.

[9] J. M.Costa and A. N. Venetsonopoulos, "Design of CIrcularly Symmetric Two-Dimensional Recursive Filters", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-22, No. 6, 1974, pp. 432-442.

APPENDIX A

Summary of Two Dimensional Digital Recursive Digital Filter Stability analysis Results

Given the two dimensionaldigital recursive filter with the corresponding biavariate recursive equation

$$g(m,n) = \sum_{J=0}^{L} \sum_{K=0}^{L} a_{JK} f(m-J,n-K) - \sum_{\substack{J=0 \\ J+K>0}}^{L} \sum_{K=0}^{L} b_{JK} g(m-J,n-K) \quad (A.1)$$

where g(m,n) is the current output, g(m-J,n-K) represents past output values and f(m-J,n-K) represents current and past input values for all permissable values of J and K. It possible to represent this relationship by a matrix recursive equation (see Appendix C) (A.2

$$G_{m,n} = \bar{B}_1 G_{m-1,n} + \bar{B}_2 G_{m,n-1} + \bar{A} F_{m,n}$$

where $G_{m,n}$ is a column vector such that all its elements are the outputs, g(m-J,n-K), where $0 \leq J \leq L$ and $\emptyset \leq K \leq L$. $F_{m,n}$ is a column vector such that its elements are the inputs, f(m-J,n-K) and B_1, B_2 and A are appropriate coefficent matrices such that (A.1) and A.2) are equivalent.

Theorem 1: Given the discrete system represented by the matrix recursive equation in (A.2). If either spectral radii, $\rho(B_1)$ or $\rho(B_2)$ is greater than or equal to one, then the system is computationally unstable.

Theorem 2: Given the discrete system represented by the matrix recursive equation in (A.2). The system is computationally unstable if $\rho(B_1+B_2)$ is greater than or equal to one.

Theorem 3: Given the discrete system represented by the matrix recursive equation in (A.2). The system is stable is

$$\rho\left[\mathrm{abs}(\bar{B}_1) + \mathrm{abs}(\bar{B}_2)\right] < 1 \quad (A.3)$$

where abs(B_1) and abs(B_2) refers to taking the absolute value of all elements in B_1 and B_2 respectively.

Theorem 4: Given the discrete system represented by the matrix recursive equation in (A.2). Define a particular permutation matrix S (See Appendix C). The system is stable if $\rho(B_1+B_2)<1$, $\rho(B_1S)<1/2$ and $\rho(B_2S)<1/2$.

Conjecture: Given the discrete system represented by the matrix recursive equation in (A.2). The system is stable if and only if $\rho(B_1)<1$, $\rho(B_2)<1$ and $\rho(B_1+B_2)<1$.

Proofs for Theorems 1 through 4 have been developed. Current research is directed toward verifying the practical usage of these theorems and to further investigation of the conjecture.

APPENDIX B

Filter Design Examples

The filter synthesis procedure for designing two dimensional digital recursive filters in this research effort is an extension of a one dimensional filter synthesis procedure. The squared magnitude characteristic of the desired circularly symmetric two dimensional filter is chosen in the Laplace Transform domain. The Butterworth filter characteristic has been chosen because of its wide spead use in band pass filter applications. The bilinear transformation is then used to map the squared magnitude characteristic into the two dimensional Z-Transform domain. The coefficient matrices, B_1 and B_2 of the corresponding matrix recursive equation (See Appendix C) are obtained and the eigenvalues of the matrix sum $(B_1 + B_2)$ are determined. These eigenvalues occur in reciprocal pairs because the original function was a magnitude response. The eigenvalues with magnitudes less than one are then used as roots of a product separable denominator to form the denominator of the two dimensional Z-Transform for a stable filter. The numerator for the filter is retained from the mapping of the squared magnitude characteristic to the two dimensional Z-Transform.

This procedure has been used to design and implement two dimensional recursive lowpass, highpass, low frequency boost and high frequency boost filters. Some examples are given below. It should be emphasized that this design procedure always results in stable filters. Also, the filter algorithm necessary to implement these filters has been developed for a CDC 6400 System by this researcher at the Naval Intelligence Support Center in Suitland, Maryland.

There are still some minor problems remaining with this procedure with

regard to obtaining circular symmetry when the critical frequency is near the Nyquist frequency or near zero. These problems are being studied with the intent of providing necessary corrective improvements in these areas. Preliminary results indicate that the problem near the Nyquist frequency is caused by the mapping of the squared magnitude characteristic to the two dimensional Z-Transform using the bilinear transformation. However, it does appear that significant correction can be made for the case when the critical frequency is near zero.

Figure B.1 shows the contour plot of a low pass filter design with a cutoff frequency of Ø.4 of the Nyquist frequency. The contour labeled D is the half power point. Figure B.2 is the contour plot of a high frequency boost filter with a break frequency of Ø.5 and relative boost of high frequency to 25.6. The contour labeled D is the half power point. Figure B.3 is The prospective plot of this filter. The design goal is that the contours and specifically the break frequency contour be circularly symmetric.

FIG. B.1: Contour Plot for Low pass Filter.
Cutoff frequency is 0.4 Nyquist Frequency

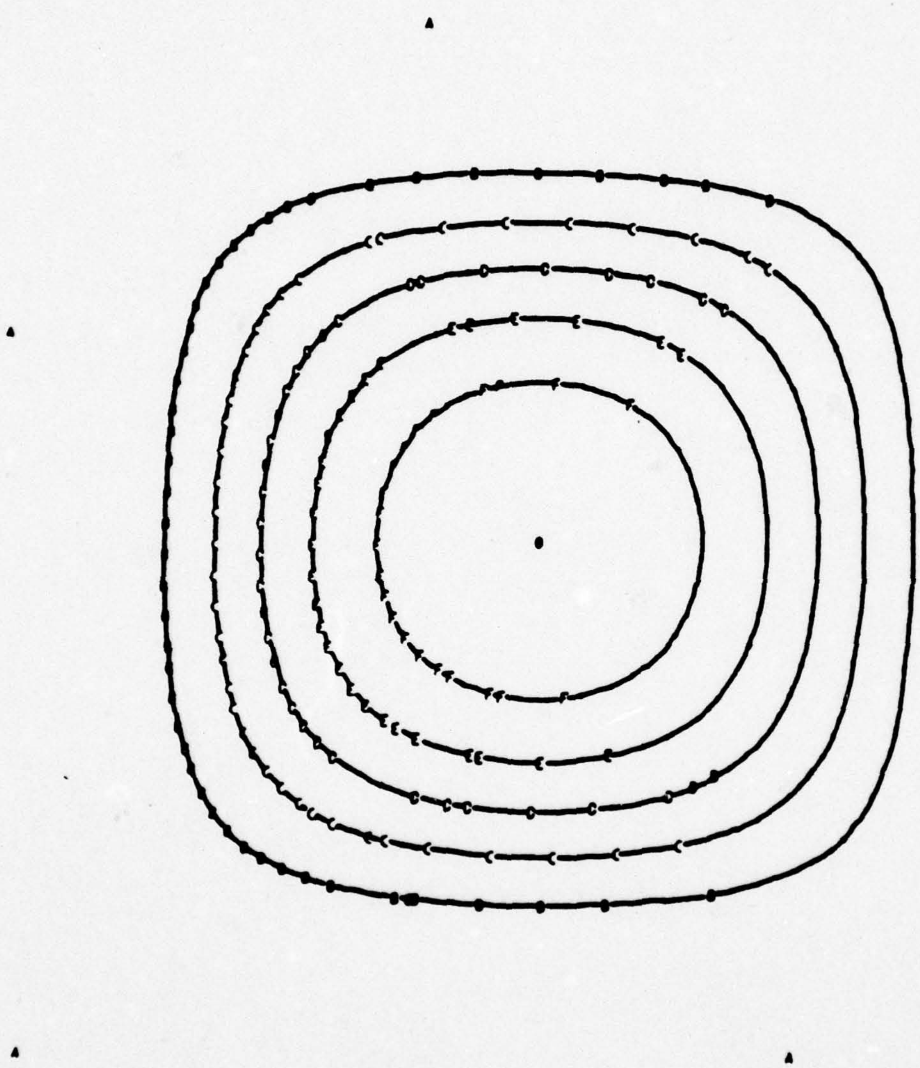


FIG. B.2: Contour Plot for high frequency boost filter.
Break frequency is 0.5 Nyquist frequency.
Relative boost of high frequencies is 25.6

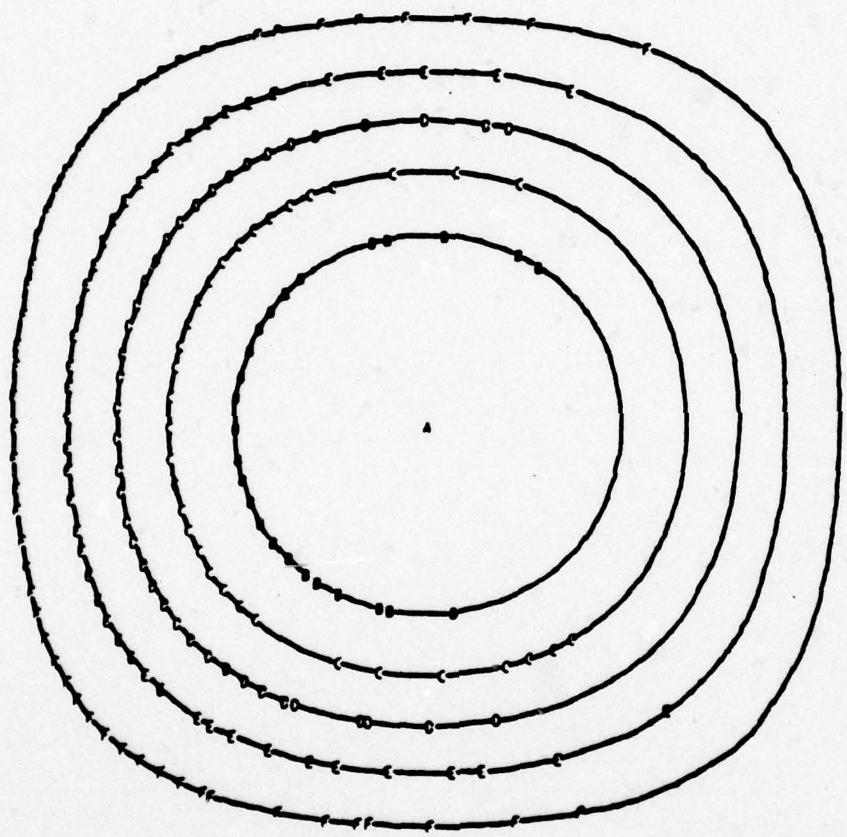
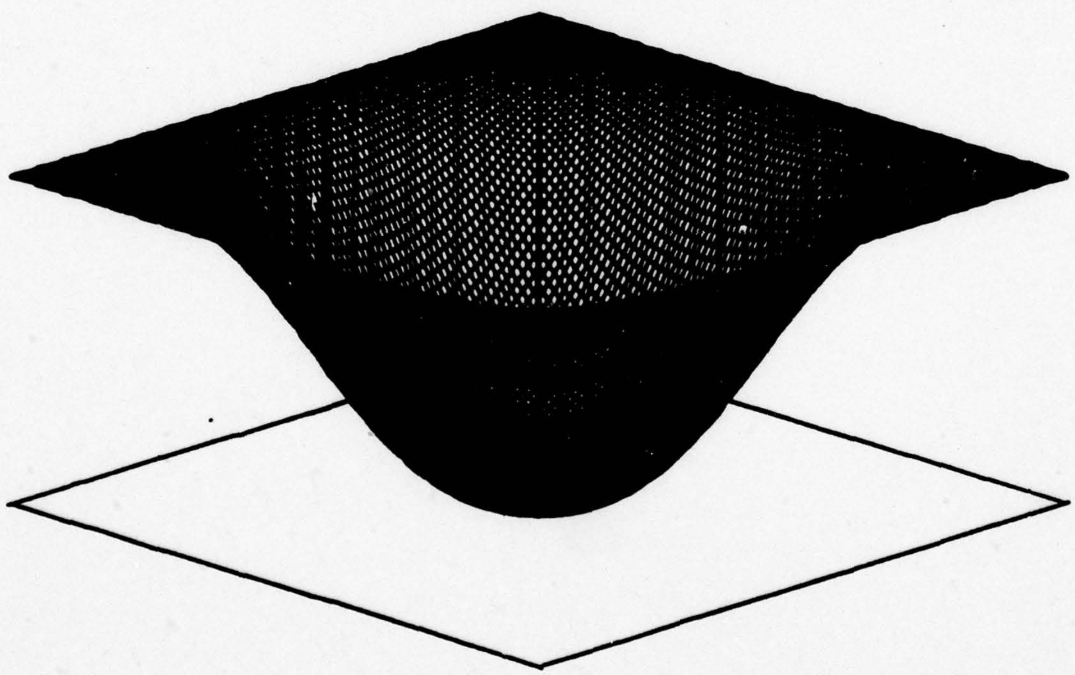


FIG.B.3: Perspective plot for high frequency boost filter of Figure B.2



APPENDIX C

Description of the Matrix Recursive Equation

C.1 Introduction

In this appendix, the matrix representation of the two dimensional digital recursive filter is presented in detail. The matrix S which is used for the Proof of Theorem 4 for stability analysis as presented in Appendix A is also described.

C.2 The Matrix Recursive Equation

Consider the two dimensional digital recursive filter which has the ZW-Transform

$$H(z,w) = \frac{\sum\limits_{J=0} \sum\limits_{K=0} a_{JK} z^{-J} w^{-K}}{\sum\limits_{J=0}^{L} \sum\limits_{K=0}^{L} b_{JK} z^{-J} w^{-K}} \tag{C.1}$$

The corresponding two dimensional recursive algorithm for the filter is given by

$$g(m,n) = \sum_{J=0}^{L} \sum_{K=0}^{L} a_{JK} f(m-J,n-K) - \sum_{\substack{J=0 \\ J+K>0}}^{L} \sum_{K=0}^{L} b_{JK} g(m-J,n-K) \tag{C.2}$$

Define the matrix V such that the element of V in the Jth row and Kth column is given by $b_{J-1,K-1}$. That is

$$V = \begin{bmatrix} b_{00} & b_{01} & \cdot & \cdot & \cdot & b_{0L} \\ b_{01} & b_{11} & \cdot & \cdot & \cdot & b_{1L} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ b_{L0} & b_{L1} & \cdot & \cdot & \cdot & b_{LL} \end{bmatrix}$$

or

$$V = \left[v_{JK} \right] = \left[b_{J-1,K=1} \right] \tag{C.4}$$

V is the filtering matrix corresponding to the denominator of the ZW-Transform H(z,w). A similar filtering matrix can be defined for the numerator of the ZW-Transform. However, this matrix is not used in this development.

Define the vector $G_{m,n}$ which contains all of the outputs in the recursive algorithm for the filter as given in (C.2). Order the outputs such that the outputs g(m-J,n), for all values of J, occur before the output g(m,n-1) and the outputs g(m-J,n-1), for all values of J, occur in order before the output g(m,n-2). Continue in this manner until all outputs are included. The vector $G_{m,n}$ is then given by

$$G = \begin{bmatrix} g(m,n) \\ g(m-1,n) \\ \vdots \\ g(m-L,n) \\ g(m,n-1) \\ \vdots \\ g(m-L,n-1) \\ \vdots \\ g(m-L,n-L) \end{bmatrix} \tag{C.5}$$

The vector $G_{m-1,n}$ is then obtained by decreasing the first parameter of each output in $G_{m,n}$ by one. That is

$$G_{m-1,n} = \begin{bmatrix} g(m-1,n) \\ g(m-2,n) \\ \vdots \\ g(m-L,n) \\ g(m-1,n-1) \\ \vdots \\ g(m-L-1,n-1) \\ \vdots \\ g(m-L-1,n-L) \end{bmatrix} \tag{C.6}$$

The vector $G_{m,n-1}$ is obtained by decreasing the second parameter of each of the outputs in $G_{m,n}$. Thus

$$G_{m,n-1} = \begin{bmatrix} g(m,n-1) \\ g(m-1,n-1) \\ \vdots \\ g(m-L,n-1) \\ g(m,n-2) \\ \vdots \\ g(m-L,n-2) \\ \vdots \\ g(m-L,n-L-1) \end{bmatrix} \tag{C.7}$$

Define $F_{m,n}$ as the vector which contains all of the inputs in the recursive algorithm for the filter as given in (C.2). Order the inputs in the same as the outputs were ordered for the vector $G_{m,n}$. Then

$$F_{m,n} = \begin{bmatrix} f(m,n) \\ f(m-1,n) \\ \vdots \\ f(m-L,n) \\ f(m,n-1) \\ \vdots \\ f(m-L,n-1) \\ \vdots \\ f(m-L,n-L) \end{bmatrix} \quad (C.8)$$

Define the matrix B_1 with elements $b_{JK}^{(1)}$ and the matrix B_2 with elements $b_{JK}^{(2)}$. B_1 and B_2 are $(L+1)^2$ by $(L+1)^2$ matrices such that

(C.9)

$$B_1 = \left[b_{JK}^{(1)}\right]$$

$$B_2 = \left[b_{JK}^{(2)}\right] \quad (C.10)$$

The elements $b_{JK}^{(1)}$ are given by the algorithm

for J = 1 to L + 1

For I = 1 to L

Let K = I + (J-1)(L+1)

If J = 1, $b_{IK}^{(1)} = -b_{I,J-1} = -V_{I+1,J}$

If J = 1, $b_{K+1,K}^{(1)} = 1$

If J > 1, $b_{IK}^{(1)} = -\frac{1}{2} b_{I,J+1} = -\frac{1}{2} V_{I+1,J}$

If J > 1, $b_{K+1,K}^{(1)} = \frac{1}{2}$

Otherwise $b_{JK}^{(1)} = 0$

The elements $b_{JK}^{(2)}$ are given by the algorithm

For J = 1 to L

For I = 1 to L + 1

Let K = I + (J-1)(L+1)

If I = 1, $b_{1K}^{(2)} = -b_{I-1,J} = -V_{I,J+1}$

If I = 1, $b_{K+L,K}^{(2)} = 1.0$

If I > 1, $b_{1K}^{(2)} = -\frac{1}{2} b_{I-1,J} = -\frac{1}{2} V_{I,J+1}$

If I > 1, $b_{K+L,K}^{(2)} = \frac{1}{2}$

Otherwise $b_{JK}^{(2)} = 0$

Define the $(L+1)^2$ by $(L+1)^2$ matrix A with elements α_{JK}. Here we depart from the standard notation to avoid confusion between the elements of the matrix A and the coefficients of the filter specified by (C.1). Then

$$A = \left[\alpha_{JK}\right] \tag{C.11}$$

The elements α_{JK} are given by the algorithm:

For I = 1 to L + 1

For J = 1 to L + 1

Let K = I + (J-1)(L+1)

$\alpha_{1,K} = a_{I-1,J-1}$

Otherwise $\alpha_{J,K} = 0$

Example C.1

COnsider the filter specified by (C.1) or (C.1) where L is equal to 2. In that case, $G_{m,n}$, $G_{m-1,n}$, $G_{m,n-1}$ and $F_{m,n}$ are 9 x 1 vectors and B_1, B_2, and A are 9 x 9 matrices. The vectors $G_{m,n}$, $G_{m-1,n}$, $G_{m,n-1}$ and $F_{m,n}$ are given by

$$G_{m,n} = \begin{bmatrix} g(m,n) \\ g(m-1,n) \\ g(m-2,n) \\ g(m,n-1) \\ g(m-1,n-1) \\ g(m-2,n-1) \\ g(m,n-2) \\ g(m-1,n-2) \\ g(m-2,n-2) \end{bmatrix} \qquad \text{(C.12)}$$

$$G_{m-1,n} = \begin{bmatrix} g(m-1,n) \\ g(m-2,n) \\ g(m-3,n) \\ g(m-1,n-1) \\ g(m-2,n-1) \\ g(m-3,n-1) \\ g(m-1,n-2) \\ g(m-2,n-2) \\ g(m-3,n-2) \end{bmatrix} \tag{C.13}$$

$$G_{m,n-1} = \begin{bmatrix} g(m,n-1) \\ g(m-1,n-1) \\ g(m-2,n-1) \\ g(m,n-2) \\ g(m-1,n-2) \\ g(m-2,n-2) \\ g(m,n-3) \\ g(m-1,n-3) \\ g(m-2,n-3) \end{bmatrix} \tag{C.14}$$

and

$$F_{m,n} = \begin{bmatrix} f(m,n) \\ f(m-1,n) \\ f(m-2,n) \\ f(m,n-1) \\ f(m-1,n-1) \\ f(m-2,n-1) \\ f(m,n-2) \\ f(m-1,n-2) \\ f(m-2,n-2) \end{bmatrix} \tag{C.15}$$

The matrices B_1 , B_2 and A are given by

$$B_1 = \begin{bmatrix} -b_{10} & -b_{20} & 0 & -\frac{1}{2}b_{11} & -\frac{1}{2}b_{21} & 0 & -\frac{1}{2}b_{12} & -\frac{1}{2}b_{22} & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \end{bmatrix} \tag{C.16}$$

$$B_2 = \begin{bmatrix} -b_{01} & -\frac{1}{2}b_{11} & -b_{21} & -b_{02} & -\frac{1}{2}b_{12} & -\frac{1}{2}b_{22} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \end{bmatrix} \tag{C.17}$$

$$A = \begin{bmatrix} a_{00} & a_{10} & a_{20} & a_{01} & a_{11} & a_{21} & a_{02} & a_{12} & a_{22} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{C.18}$$

We can then express (C.2) in the matrix form

$$G_{m,n} = B_1 G_{m-1,n} + B_2 G_{m,n-1} + AF_{m,n} \tag{C.19}$$

Note that if we write the equations corresponding to the rows of (C.19) we obtain for the first row

$$g(m,n) = -\sum_{J=0}^{L} \sum_{K=0}^{L} b_{JK} g(m-J,n-K) + \sum_{\substack{J=0 \\ J+K>0}}^{L} \sum_{K=0}^{L} a_{JK} f(m-J,n-K) \quad (C.20)$$

For the subsequent rows, we obtain

$$g(m-J,n-K) = g(m-J,n-K) \quad (C.21)$$

or the outputs are equated to themselves. It follows directly that (C.19) is equivalent to (C.2).

C.3 The S Matrix

We now give the algorithm for the matrix which is used to reorder the rows and columns of the matrices B_1 and B_2 for the proof of Theorem 4 as described in Appendix A. Define the $(L+1)^2$ by $(L+1)^2$ matrix S such that

$$S = \left[s_{JK}\right] \quad (C.22)$$

The elements S_{JK} are given by the algorithm

For I = 1 to L + 1

For M = 1 to L + 1

Let J = M + (I-1)(L+1)

Let K = I + (M-1)(L+1)

$s_{JK} = 1.0$

Otherwise $s \quad = 0$

Example C.2

Consider the filter specified by (C.1) or (C.2) where L is equal to 2.

In that case, we have

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

APPENDIX D

Stability Analysis of Two-Dimensional Recursive Filters
(A Preprint)

Stability Analysis of Two-Dimensional Recursive Filters*

Winser E. Alexander, Member, IEEE

and

Steven A. Pruess

---

* Manuscript received October 10, 1975; revised January 31, 1977 and June 23, 1978. This work was partially supported by the U. S. Department of Energy and partially by the Naval Air Systems Command.

W. E. Alexander is with the Department of Electrical Engineering at North Carolina Agricultural and Technical State University

S. A. Pruess is with the Department of Mathematics at the University of New Mexico.

ABSTRACT

A new approach to the stability problem for the two-dimensional digital recursive filter is presented. The bivariate difference equation representation of the two-dimensional recursive filter is converted to a multi-input multi-output (MIMO) system similar to the state space representation of the one dimensional digital recursive filter. In this paper, a pseudo-state space representation is used and three two-dimensional polynomial matrices are obtained. A general theorem for stability of two-dimensional digital recursive filters is derived and a very useful theorem is presented which expresses sufficient requirements for instability in terms of the spectral radii of these matrices.

I. Introduction

A two-dimensional digital recursive filter can be characterized by the bivariate difference equation

$$g(m,n) = \sum_{J=0}^{L} \sum_{K=0}^{L} a_{JK} f(m-J,n-K) - \sum_{J=0}^{L} \sum_{K=0}^{L} b_{JK} g(m-J,n-K) \qquad (1)$$

where the coefficients a_{JK} and b_{JK} are constants [1] and some of these constants may be zero. There are two major problems to consider in the design of recursive filters for two-dimensional signal processing: synthesis and stability. The synthesis problem consists of determining the filter coefficients so that the required frequency response is realized. If the resulting filter is to be useful, it must be bounded input-bounded output (BIBO) stable. In this paper the stability problem is considered and a new approach to stability analysis for the two-dimensional digital recursive filter is presented.

For the one-dimensional case, there are essentially two methods of determining necassary and sufficient conditions for stability of digital filters: examining regions of analyticity for the characteristic polynomial and by direct evaluation of the characteristics of the impulse response [2,3,4]. In particular, if the system corresponding to the digital filter is represented by a state space equation, then one can determine stability from the coefficient matrices in the state space equation[4]. For the two-dimensional case, generalizations of the first method involves examining regions of analyticity for bivariate polynomials which is computationally feasible only for very simple filters[5]. This paper attempts to generalize the second method for the two-dimensional case, i.e. to establish stability by computing the spectral radii of coefficient matrices with real coefficients.

II. Pseudo State Space Representation

Fornasini and Marchesini [6] have defined a state space representation of the two-dimensional digital recursive filter. In this paper, we use a particular case of the Fornasini-Marchesini model where one of the coefficient matrices is the null matrix. Thus, we obtain the pseudo state space representation

$$G_{m,n} = B_1 G_{m-1,n} + B_2 G_{m,n-1} + AF_{m,n}$$
$$g(m,n) = DG_{m,n} \tag{2}$$

$G_{m,n}$ is a column vector such that its elements are the outputs, $g(m-J,n-K)$ where $\emptyset \leq J \leq L$ and $\emptyset \leq K \leq L$. Note that $G_{m,n}$ contains all of the outputs that are represented in (1) including $g(m,n)$. Similarly, $F_{m,n}$ is a column vector such that its elements are the inputs, $f(m-J,n-K)$ where $\emptyset \leq J \leq L$ and $\emptyset \leq K \leq L$.

We can then define matrices B_1, B_2 and A [7] such that (1) and (2) are equivalent. The matrices B_1, B_2 and A are all of order $(L+1)^2$ by $(L+1)^2$. The vector D is a row vector with L+1 elements.

The ordering of the outputs in $G_{m,n}$ and of the inputs in $F_{m,n}$ is not unique. However, the ordering does affect the relative position of the elements of the corresponding coefficient matrices. Also note that there are identical elements in $G_{m-1,n}$ and $G_{m,n-1}$. Where this occurs, the corresponding elements of B_1 and B_2 can be divided such that the magnitude of each is no larger than that of the corresponding b_{JK} or one as appropriate. It is convenient to consistently divide equally and choose a particular ordering scheme.

III. Stability Analysis

The stability analysis herein will be confined to the linear shift invariant (LSI) two-dimensional discrete system. Such a system is BIBO stable if and only if the discrete impulse response of the system, $h(m,n)$, is absolutely summable, i.e., $\sum_{m,n=0}^{\infty} |h(m,n)| < \infty$ [1].

Let us define the particular vector $H_{J,K}$ as that input vector which represents a single unit sample at the (J,K) position of the two-dimensional data array and all other inputs are zero. Let us further define the initial condition vector, $G_{J-1,K}$ and $G_{J,K-1}$, as null vectors. Then for m=J and n=K, (2) reduces to

$$\begin{aligned} G_{J,K} &= AH_{J,K} \\ h(J,K) &= DG_{J,K} \end{aligned} \tag{3}$$

Define the term $C(B_1^J, B_2^K)$ as the sum of all unique products involving B as a factor J times and B_2 as a factor K times. It is helpful to note that if B_1 and B_2 commute, then $C(B_1^J, B_2^K) = \binom{J+K}{K} B_1^J B_2^K = (J+K)! B_1^J B_2^K / (J!K!)$. In general, the matrices do not commute. Therefore, we give as an example $C(B_1^2, B_2^1) = B_1^2 B_2 + B_1 B_2 B_1 + B_2 B_1^2$.

Lemma 1: Given the discrete LSI system represented by (2), the contribution to the output vector, $G_{m,n}$, by a single input vector, $H_{J,K}$, which corresponds to a unit impulse at the (J,K) position where $J \leq M$ and $K \leq N$ is given by $G_{m,n} = C(B_1^{m-J}, B_2^{n-K}) AH_{J,K}$.

The proof of Lemma 1 is given in the Appendix. Lemma 1 provides a convenient means of finding the output of the two-diemnsional digital recursive filter for all values of m and n when the filter is excited by a single input at any point in the array. Since the filter is linear and shift invariant, we can use the principle of superposition to find the output for

any particular sequence of inputs.

Thus, the unit impulse response of the filter is given by

$$G_{m,n} = C(B_1^m, B_2^n)AH_{0,0}$$

$$h(m,n) = DG_{m,n} = DC(B_1^m, B_2^n)AH_{0,0} \tag{4}$$

Lemma 2: Given the discrete LSI system represented by (2) for which the corresponding transfer function has mutually prime numerator and denominator polynomials. If the contribution to the output vector $G_{m,n}$ by a bounded sequence of input vectors $F_{J,K}$ where $\emptyset \le J \le M$ and $\emptyset \le K \le N$ can be expressed by $G_{m,n} = Q^m AF_{J,K}$ or $G_{m,n} = Q^n AF_{J,K}$, then the system is unstable if $\rho(Q)$, the spectral radius of Q, is greater than one. The proof of Lemma 2 is given in the Appendix.

Theorem 1: The discrete LSI system represented by (2) is stable if and only if for at least one matrix norm

$$S_2 = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \left\| DC(B_1^m , B_2^n)AH_{0,0} \right\| < \infty$$

Theorem 1 follows directly from (4) and the requirement that the discrete impulse response be absolutely summable. Since $h(m,n)$ is a scaler, its matrix norm is equivalent to its absolute value and the proof of theorem 1 is obvious.

Theorem 2: The discrete LSI system represented by (2) and for which the numerator and denominator polynomials of the corresponding transfer function are mutually prime is unstable if any one of the spectral radii $\rho(B_1)$, $\rho(B_2)$ or $\rho(B_1+B_2)$ is greater than or equal to one. The proof of Theorem 2 is given in the Appendix.

In the practical application of two-dimensional digital recursive filters,

any filter with $\rho(B_1)$, $\rho(B_2)$ or $\rho(B_1+B_2)$ equal to one can be considered to be unstable and should be avoided[8]. Goodman [5] has shown by clever examples that two-dimensional filters with nonessential singularities of the second kind on the unit bidisc may be stable. Such a filter may have $\rho(B_1)$, $\rho(B_2)$ or $\rho(B_1+B_2)$ equal to one. However, roundoff errors and coefficient truncation would prevent satisfactory performance by such a filter for most applications.

Several other theorems relating to sufficient conditions for stability have been found [7]. However, it has been shown that these constraints are too restrictive for general use. That is, useful stable filters can be found which do not satisfy the corresponding sufficient conditions for stability.

Computer algorithms are readily available to find the spectral radius of a matrix with real coefficients. Thus, Theorem 2 presents a convenient and easily implemented technique to assess the stability of two-dimensional digital recursive filters.

APPENDIX

In this appendix, the proofs for Lemmas 1 and 2 and Theorem 2 are given. Since all vector norms are equivalent, any convenient norm may be used for either input or output.

A1. Proof of Lemma 1

We proceed with a proof by induction. If we use (2) and (3) to obtain $G_{J+1,K}$, $G_{J,K+1}$ and $G_{J+1,K+1}$ for input vector $H_{J,K}$ and all initial condition vectors are null vectors, we obtain

$$\begin{aligned} G_{J+1,K} + B_1 G_{J,K} &= B_1 A H_{J,K} \\ G_{J,K+1} = B_2 G_{J,K} &= B_2 A H_{J,K} \\ G_{J+1,K+1} = B_1 G_{H,K+1} + B_2 G_{K+1,J} &= (B_1 B_2 + B_2 B_1) A H_{J,K} \end{aligned} \tag{A1}$$

If we use Lemma 1, we obtain

$$\begin{aligned} G_{J+1,K} &= C(B_1^0, B_2^1) A H_{J,K} = B_1 A H_{J,K} \\ G_{J,K+1} &= C(B_1^0, B_2^1) A H_{J,K} = B_2 A H_{J,K} \\ G_{J+1,K+1} &= C(B_2^1, B_2^1) A H_{J,K} = (B_1 B_2 + B_2 B_1) A H_{J,K} \end{aligned} \tag{A2}$$

Thus for any arbitrary m and n such that m>J and n>K, we can use (2) to write

$$G_{m+1,n} = B_1 G_{m,n} + B_2 G_{m+1,n-1} \tag{A3}$$

Then using (4) to find expressions for $G_{m,n}$ and $G_{m+1,n-1}$, we have

$$G_{m+1,n} = [B_1 C(B_1^{m-J}, B_2^{n-J}) + B_2 C(B_2^{m-J+1}, B_2^{n-K-1})] A H_{J,K} \tag{A4}$$

Consider the term, $C(B_1^J, B_2^K)$. All of the products in the term either have B_1 as the first factor or B_2 as the first factor. If B_1 is the first factor we must postmultiply by the sum of all possible products such that the power of B_1 is decreased by one. If B_2 occurs as the first factor, we must postmultiply by the sum all possible products such that the power of B_2 if decreased by one. We conclude that

$$C(B_1^J, B_2^K) = B_1 C(B_1^{J-1}, B_2^K) + B_2 C(B_1, B_2^{K-1}), \qquad \text{(A5)}$$

for all J and K such that both J and K are greater than or equal to one. It follows directly that

$$G_{m+1,n} = C(B_1^{m+1-J}, B_2^{n-K}) AH_{J,K} \ . \qquad \text{(A6)}$$

Similarly from (2) we write

$$G_{m,n+1} = B_1 G_{m-1,n} + B_2 G_{m,n} \ . \qquad \text{(A7)}$$

Using (4) to find expressions for $G_{m-1,n+1}$ and $G_{m,n}$, we have

$$G_{m,n+1} = [B_1 C(B_1^{m-J-1}, B_2^{n+1-K}) + B_2 C(B_1^{m-J}, B_2^{n-K})] AH_{J,K} . \qquad \text{(A8)}$$

It follows that

$$G_{m,n+1} = C(B_1^{m-J}, B_2^{n+1-K}) AH_{J,K} \qquad \text{(A9)}$$

Finally, from (2) we obtain

$$G_{m+1,n+1} = B_1 G_{m,n+1} + B_2 G_{m+1,n} \qquad \text{(A10)}$$

Using Lemma 1 to express $G_{m,n+1}$ and $G_{m+1,n}$ we obtain

$$G_{m+1,n+1} = [B_1 C(B_1^{m-J}, B_2^{n+1-K}) + B_2\, C(B_1^{m+1-J}, B_2^{n-K})] AH_{J,K} \qquad \text{(A11)}$$

It follows from (A5) and (A11) that

$$G_{m+1,n+1} = C(B_1^{m+1-J}, B_2^{n+1-K}) AH_{J,K} \qquad \text{(A12)}$$

and Lemma 1 holds.

A2. Proof of Lemma 2

In the proof of Lemma 2, we shall show that if the response to a particular sequence of input vectors can be represented as given in Lemma 2, then the system is unstable if $\rho(Q)>1$ [9].

Define the eigenvalue corresponding to the spectral radius of Q as and the corresponding eigenvector as P_Q. Then if the system transfer function has mutually prime numerator and denominator polynomials we can select an input vector such that

$$AF_{J,K} = \varepsilon P_Q + R_{J,K} \text{ for all J and K} \tag{A13}$$

where ε is an arbitrary nonzero finite constant and $R_{J,K}$ is not in the direction of P_Q. We then have

$$G_{m,n} = Q^m AF_{J,K} = \varepsilon Q^m P_Q + Q^m R_{J,K} \tag{A14}$$

Then since λ_Q is the eigenvalue corresponding to the spectral radius, the norm of $G_{m,n}$ is dominated by the term εQP_Q in the limit as m approaches infinity.

Thus

$$S = \lim_{m\to\infty} \left\| G_{m,n} \right\| = \text{Lim} \left\| \varepsilon Q^m P_Q \right\| = \lim_{m\to\infty} \left\| \varepsilon \lambda_Q{}^m P_Q \right\|$$

Note that S is infinite if λ_Q is greater than one and Lemma 2 holds.

A3. Proof of Theorem 2

For this proof, we show that we can find a particular sequence of inputs that give unbounded output if either of the spectral radii specified in Theorem 2 is greater than one.

From Lemma 1 and 2 the output from a single arbitrary bounded input at the (J,K) position can be given by

$$G_{M,N} = f(J,K)C(B_1^{M-J}, B_2^{N-K})AH_{J,K} \tag{A16}$$

$$g(M,N) = DG_{M,N}$$

where F(J,K) is the scalar input at the (J,K) position. If we let K=N and J=Ø in (A16), we have

$$G_{M,N} = F(O,N)C(B_1^M, B_2^0)AH_{J,K} = f(O,N)B_1^M AH_{J,K} \tag{A17}$$

If we apply Lemma 2, we see that the system is unstable $\rho(B_1) > 1$. If we let J = M and K = O in (A16), we have

$$G_{M,N} = f(M,0)C(B_1^0, B_2^N)AH_{J,K} = f(M,0)B_2^N AH_{J,K} \tag{A18}$$

If we apply Lemma 2, we see that the system is unstable if $\rho(B_2) > 1$.

If we use a particular sequence of inputs f(J,M-J) for $\emptyset \leq J \leq M$ where all f(J,M-J) are bounded and equal. Using the principle of superposition and (A16) we have

$$G_{M,N} = \sum_{J=0}^{M} f(J,M-J)C(B_1^{M-J}, B_2^J)AH_{J,M-J} \tag{A19}$$

Since all inputs are equal, we can write

$$G_{M,N} = f(O,M) \left[\sum_{J=0}^{M} C(B_1^{M-J}, B_1^J)\right]AH_{0,M} \tag{A20}$$

$$G_{M,M} = f(0,M) (B_1 + B_2)^M AH_{0,M} \tag{A21}$$

If we apply Lemma 2, we see that the system is unstable if $\rho(B_1 + B_2) > 1$.

ACKNOWLEDGEMENT

The author wishes to thank Professor E. I. Jury of the Department of

Electrical Engineering, University of California Berkley and the reviewers of this paper for supplying helpful comments and references.

REFERENCES

[1] Lawrence R. Rabiner and Bernard Gold, Theory and Application of Digital Signal Processing, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1975, pp. 442-455.

[2] Samuel Stearns, Digital Signal Analysis, Hayden Publishing Company, 1975, p. 134.

[3] E. I. Jury, "Theory and Application of Inners", Proc. IEEE, Vol. 63 No. 7, 1975, pp. 1044-1068.

[4] Katsuhiko Ogata, State Space analysis of Control Systems, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1967, p. 487.

[5] Dennis Goodman,"Some Stability Properties of Two-Dimensional Linear Shift-Invariant Digital Filters", IEEE Transactions on Circuits and Systems, Vol CAS24, No. 4, 1977, pp. 201-208.

[6] E. Fornasini and G. Marchesini, "State Space Realization Theory for Two-Dimensional Filters", IEEE Transactions on Automatic Control, Vol. AU 197, pp. 484-492.

[7] Winser E. Alexander, Stability and Synthesis of Two-Dimensional Digital Recursive Filters, Ph.D. dissertation, University of New mexico, Albuquerque, N. M., 1974 (University Microfilms, ann Arbor, Mich.).

[8] N. K. Bose, "Problems and Progress in Multidimensional System Theory", Proc. IEEE, Vol. 65, No. 6, 1977, pp. 824-840.

[9] Alston S. Householder, The Theory of Matrices in Numerical Analysis, Blaisdell Publishing Company, New York, N. Y., 1964, Chapter 2.

INDENTER HARDNESS STUDIES

by

George J. Filatovs

TABLE OF CONTENTS

INTRODUCTION

Technology would be much advantaged by the availability of better methods for anticipating and preventing the mechanical failure of ceramics. In particular, the dependence of mechanical properties on the microstructural character is so complex, and ceramic microstructures so inherently variable, that the technological use of ceramics has been persistently retarded.

The field of mechanical properties is vast, therefore this phase addressed itself to a well defined and limited objective; to acquire, through literature reviews and laboratory testing, and understanding of the principles of microhardness testing of ceramics. This understanding was then used to devise a modification of the microhardness test which allowed the extraction of additional information.

The isolation of the hardness test as the centerpiece for this task can be easily defended. Hardness tests have long been used for classification and surveying of various materials, and their application to ceramics promises to be useful, for example to determine fracture toughness. And in a general way, indentation serves as a model system for studying the strength degradation in a wide range of phenomena such as machining flaws and impact abrasion.

HARDNESS TESTING

The hardness test has long suggested itself as a simple means of obtaining mechanical properties: the ease of the test and the small sample size needed being the principal inducements. As a result, a number of technically valuable correlations have been established, for example, the tensile strength-Brinell hardness formula for certain metals. Unfortunately, the hardness test has not become a routine tool for investigating ceramic materials. The principal factors in this was the controversy in the relative roles of plastic deformation and cracking, (it was not until the 70's that plasticity was accepted as an important process), and the lack of theoretical underpinning to sort out confusing results.

The mechanics of point indentation has been slowly developing. As early as 1881 Hertz[1] analyzed the general elastic contact between two curved bodies, and in 1885 Boussinesq[2] solved the stress field for the case of an infinitely sharp indentor on a flat surface. However, as the stress field produced in non-homogeneous, anisotropic crystalline materials is complex, and there are no solutions for stress fields in terms of general geometry for an elastic indentor, it seems unlikely that any solution will emerge. Anyone who has made hardness tests on ceramics will have been impressed with the diversity of results which arise from minor variations in test conditions.

Nevertheless, a number of models have been devised to understand the indentation of actual materials, containing drastic simplications and based on systematic studies of ceramic materials[3,4]. Most of

these models have in common the inclusion of elastic-plastic processes, and the assumption of spherical symmetry; in addition, the initiation and propagation of cracks are usually treated as separate events. There are also two extreme categories of contact situations; blunt and sharp. For blunt indentors[5], the crack nucleates from pre-existing flaws and develops into a Hertzian cone, while sharp indentors[6] nucleate cracks from the plastic zone at the contact, which then develops into half-penny cracks.

Recent theoretical and experimental results have attempted to establish the actual macroscopic events in indentation fracture. The post-mortem examination of fracture surfaces and indentation impressions have been important in directing theoretical attempts. Perhaps the most dramatic development is the apparent correlation between fracture mirror patterns and fracture stress for glass and ceramics[7-12]. Unfortunately, after much intense study and confirmation of the general empirical relationships, it appears that these relationships may have no useful fundamental implications[13-15].

Not withstanding such analytical limitations, a number of studies have established certain common features. The greatest attention has been directed to the blunt indentor; the crack systems, their nucleation, propagation, and geometry have been studied[16-21]. Generally, the blunt indentor probes the cleavage tendencies. The effect of loads[22] and indentor angle (sharpness)[3,6] have been studied. The theoretical understanding of the crack growth in these tests has been primarily based on the Griffith energy criterion.

The sharp indentor is probably more pertinent to actual contact

situations, such as grinding and hardness testing. The principal difference from the blunt indentor is the plastic flow preceeding the formation of the crack and the complex stress field. Once the crack system develops, the influence of indentor geometry becomes less important. Specific discussion of the sharp indentation studies will be in terms of the hardness test, which will be considered a semi-sharp indentor.

In summary, while there have been determined various functional relationships for indentation processes, the theoretical description remains incomplete.

HARDNESS TESTING OF CERAMICS

In spite of the formidable theoretical and experimental difficulties a number of attempts have been made to use the hardness test on ceramics. Most of these have involved the port-mortem examination of crack patterns and dimensions, and the indentor has most frequently been the Vickers diamond pyramid. This indentor is usually considered a semi-sharp indentor and its indentation stress field is favorable for plastic flow. Specific examples are the measurement of stresses in tempered glass surfaces(23). fracture toughness determination(24), and compressive strengths(25). The results of some of these will be referred to in the discussion of experimental results.

EXPERIMENTAL CONJECTURES

The experimental idea developed here is an extension of the attempts to use the features of the hardness impression such as the extent and pattern of cracking. The surface crack pattern as a clue to fracture toughness was originated by Palmquist in 1957(27) and since has been considerably refined in theory and procedure(28-29). The shortcoming

of these methods is that measurement of the crack features is difficult and unreliable, usually requiring SEM and etching techniques[30-32]. A simplier and less ambiguous method is desirable.

Although there is no solution of the stress field for hardness indentors, the deformation-fracture for well developed cracks in brittle materials is known, and can be used to orient our thoughts. Fig. 1 shows a Vickers Diamond Pyramid Indentor (DPI) and the primary crack pattern. The sharp edges of the indentor tend to initiate and develop half-penny cracks along the diagonals. Therefore, to prove the extent of these cracks it seems reasonable to search for some interaction of the cracks with stress singularities such as a surface or another indentation. For example, as shown in Fig. 2, for the orientation of indentor shown, the cracks would extend to the surface, and the crack length would be unambiguously revealed when breakthrough at the edge occurred. Another possibility is shown in Fig. 3, where the distance between successive impressions is varied until chipping occurs.

Figure 1. Diamond Pyramid Indentation and Resulting Crack Pattern

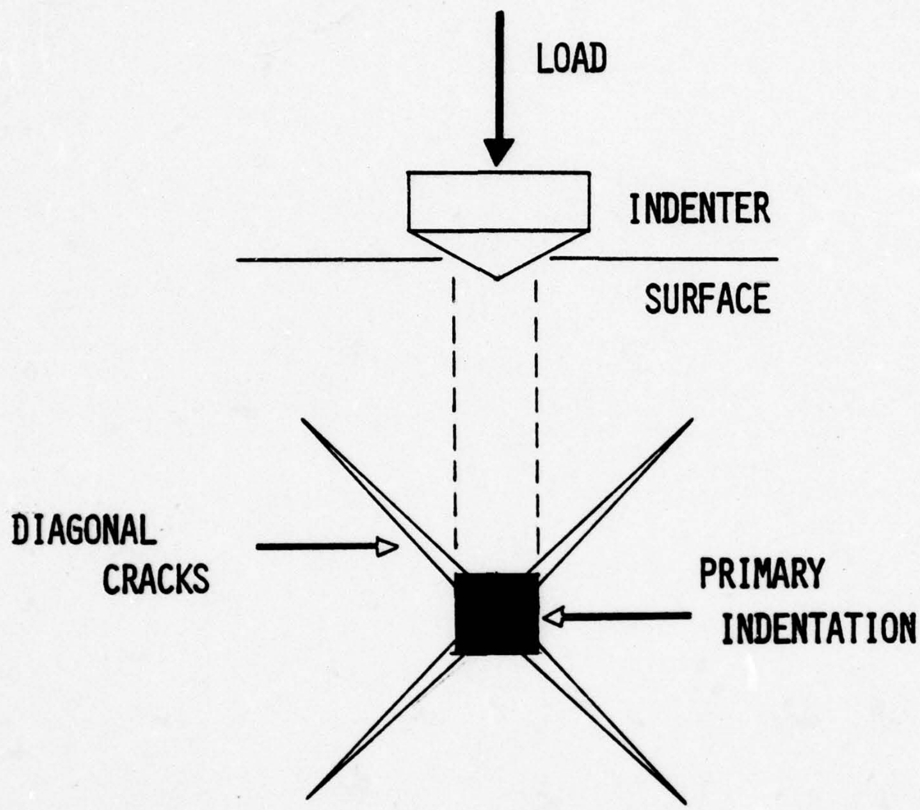
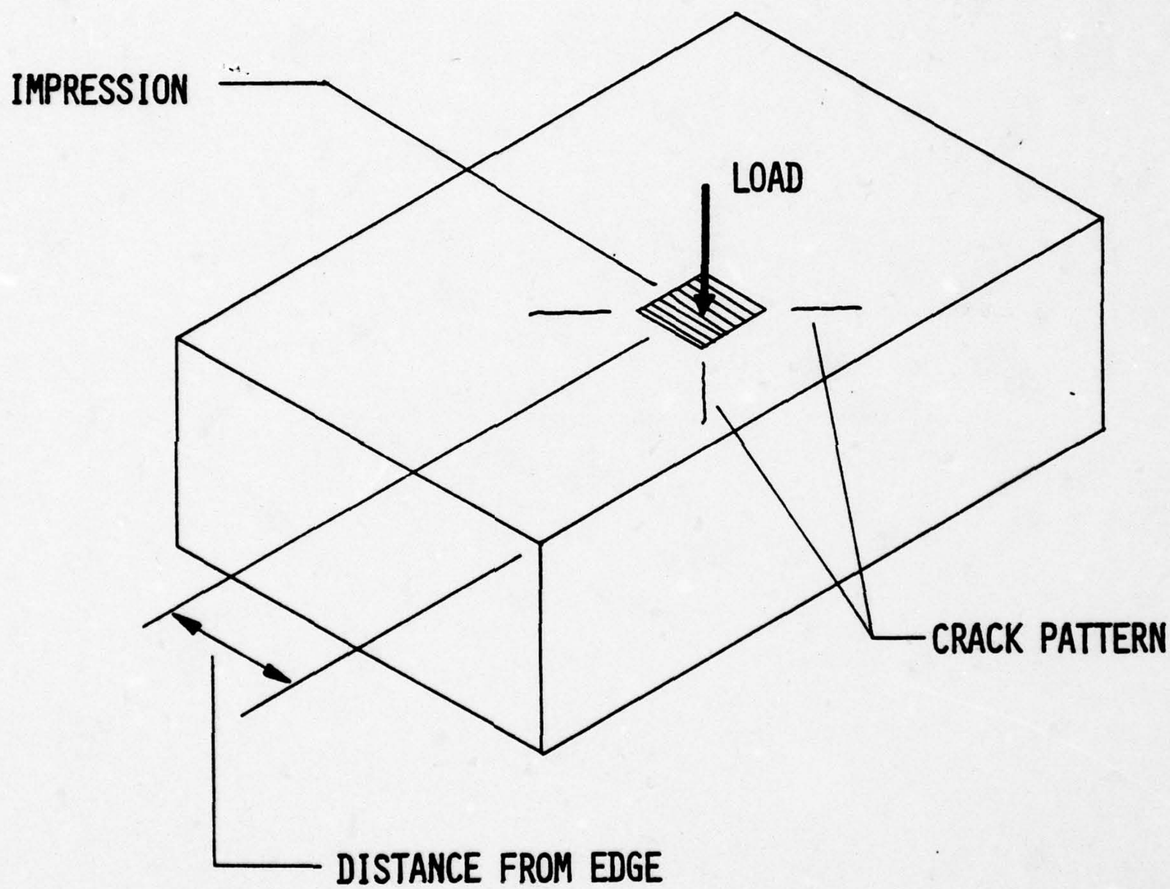


Figure 2. Diamond Pyramid Indentation made close to an edge. Cracks break out to the perpendicular surface and cause chipping.



EXPERIMENTAL MATERIAL

The materials used were remnants from an evaluation program on ceramic vane materials conducted by IIT Research Institute and the Air Force Materials Laboratory and were extensively characterized by them[33-35]. The materials are briefly described below, and some properties summarized in Table 1. For testing by North Carolina A & T State University, the surfaces were polished with diamond paste, final polishing being with 1 - μ diamond.

NC-350: Reaction bonded Si_3N_4. The microstructure has 25% porosity, with the porosity and silicon nitride phase uniformly distributed. The porosity was one-half to one-third open. The phases present were α -Si_3N_4 (major) and β -Si_3N_4(minor).

NC-435: Siliconized SiC. A two-phase material with about 20% Silicon. The SiC phase is the α -form, and low porosity which is mostly closed.

NC-132: Hot pressed Si_3N_4. The phases present were α -Si_3N_4, the major phase, and Si_2N_4O, the minor phase. There were also traces of WC. The microstructure showed fine, elongated grains with virtually no open porosity.

EXPERIMENTAL APPARATUS

The tests were conducted on an Kentron microhardness tester, using a 136° Diamond Pyramid Indentor. This indentor is cut in the shape of a square-based pyramid having an apex angle of 136°. The loads were applied for 15 seconds, and care was taken to reduce machine vibration. The tests were

conducted in a laboratory environment at room temperature. Considerable practice was initially expended in learning to obtain consistent results.

Figure 3. Expected chipping between two correctly oriented Diamond Pyramid Indentations.

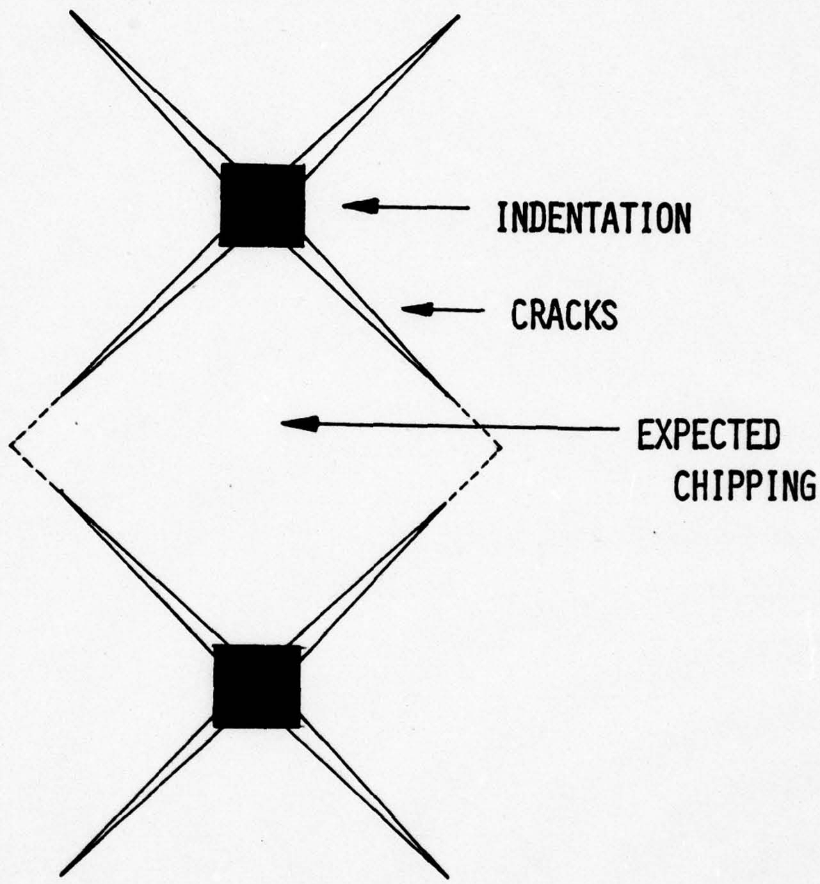


Table 1

Material	Batch	Fracture Stress	Modulus	Density
NC-350	1	30,860 psi	25.5x10 psi	2.52 gm/cc
	2	23,530	23.5	2.40
	3	33,960	27.7	2.54
NC-435	1	50,460	53.9	2.936
	3	65,980	49.5	2.997
	4	55,150	48.8	2.962
NC-132	1	90,910	48.4	3.177
	2	115,210	45.7	3.186

RESULTS

Based on the previous speculations, a number of different indentation procedures were tried. The two-indentor method did yield some success; however, the experimental procedure proved troublesome. At least on the machine used, it proved difficult to line up the impression accurately, and so not enough usable data resulted. The test using an edge, shown in Fig. 3, gave better results. In this test, the distance from an edge which caused breaking of the edge at a particular load, was proved. The results are given in Tables 2 and 3. No results are included for NC-132, as this material frequently deformed by flaking rather than forming an impression and cracking. A few usable data points indicated that the results for this material followed the trends set by the other materials. This material is included because it apparently represents the upper strength limits for materials whish can be used for the distance from the edge test, although anisotropy might be playing a role.

DISCUSSION

The data of Table 2 are plotted in Fig. 4 is the fracture stress as determined by four point bending from Table 1. While there is no single straight line which appears to fit this data well, it is possible that a line can be fitted through each of the sets of points for the two materials. Certainly the hardness test appeared to rank the materials correctly, both as to batch and type; this certainly indicates some accuracy and shows that surface finish, machine operation and microstructural variability are not scattering the data beyond use.

The data from Table 3 are shown plotted on log-log paper in Fig. 5. Comparison of this data with that in Table 1 shows that the materials are

Table 2

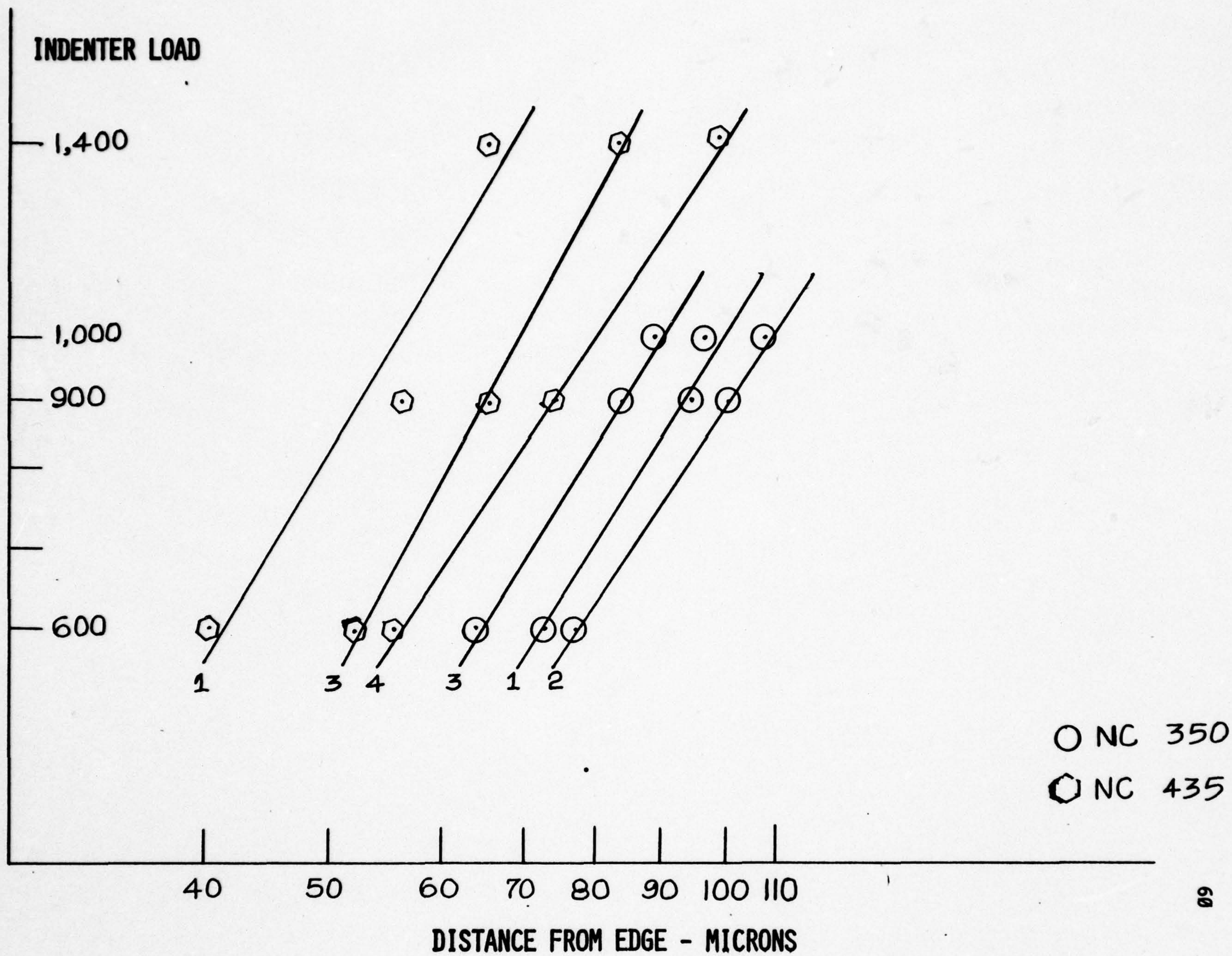
Material	Batch	Hardness*
NC-350	1	859 DPN
	2	662
	3	1087
NC-435	1	1086
	3	1780
	4	1646

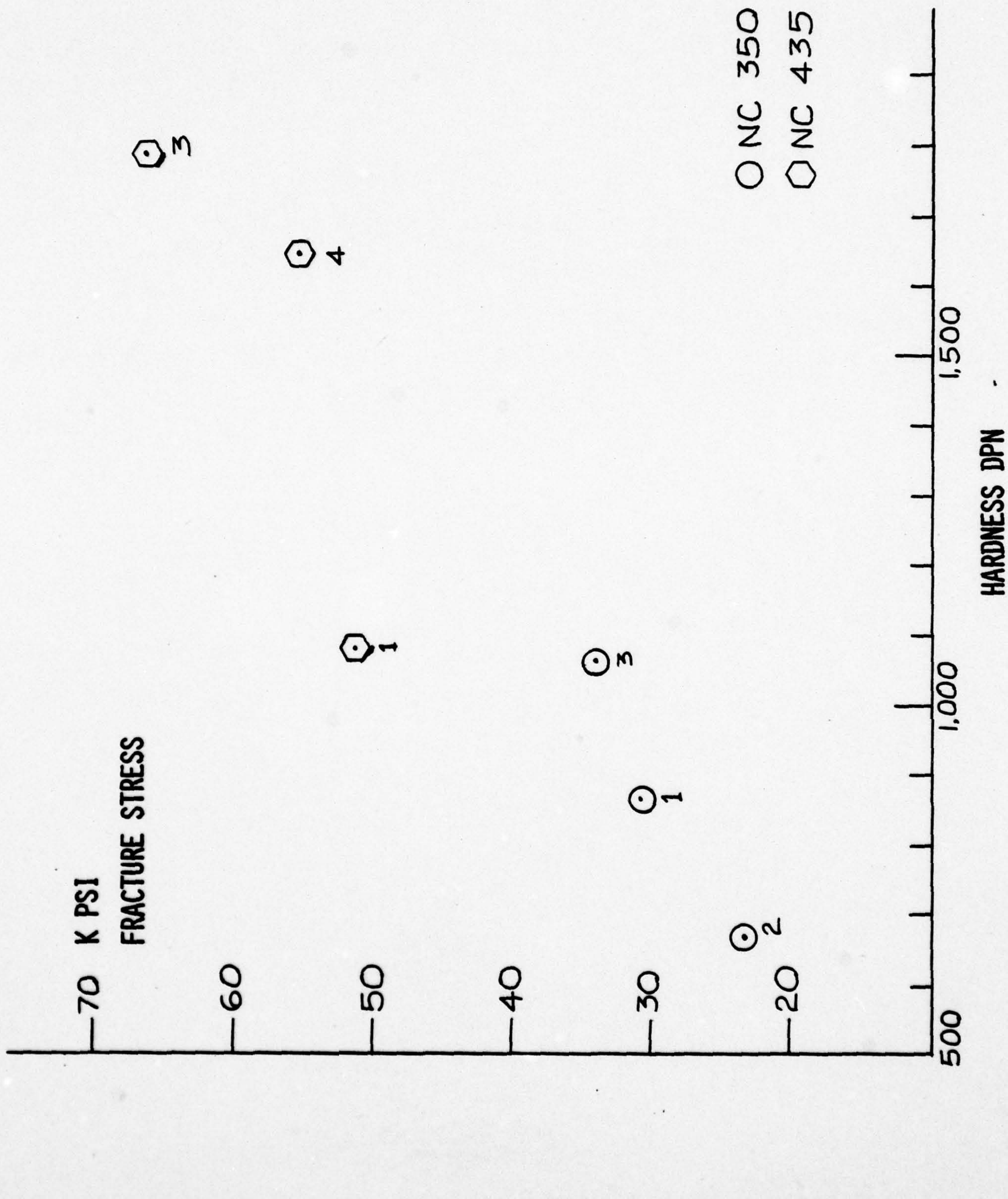
*Average of 10 readings, 600 gram load.

Table 3

Material	Batch	Load	Distance from Edge*
NC-350	1	600 grams	72 microns
	2		76
	3		64
NC-435	1	600	39
	3		52
	4		56
NC-350	1	900	94
	2		99
	3		83
NC-435	1	900	56
	3		65
	4		74
NC-350	1	1000	196
	2		105
	3		88
NC-435	1	1400	65
	3		82
	4		97

*Average of 10 readings.


INDENTER LOAD
1,400
1,000
900
600
1
3
4
3
1
2
NC 350
NC 435
40
50
60
70
80
90
100
110
DISTANCE FROM EDGE - MICRONS


K PSI
FRACTURE STRESS
70
60
50
40
30
20
500
1,000
1,500
HARDNESS DPN
NC 350
NC 435
1
2
3
1
3
4

ranked by modulus rather than by strength. In trying to understand this and other features about this data, we need some theoretical underpinning. The first obvious correlation to look for is some analogy to the mirror boundary relationship[9,10,14]

$$\nabla_f r_m^{\frac{1}{2}} = A$$

where ∇_f is the fracture stress, r_m the fracture mirror radius, and A the mirror constant. Although, as mentioned previously, the theoretical basis for this relationship has been denied, it has nevertheless been confirmed by many studies, and so there is some expection that the present data would fit this equation. If the distance from the edge is taken as an indication of the mirror size, then the functional dependence in the above equation must be reversed. That is, the mirror boundary represents the surface created by the fracture stress, so that a strong material would create a large fracture, while in the present test, as the load is constant (or normalized), the stronger material will break closer to the edge and thus create a smaller area.

Even though the functional dependence is correct, the data slope is not the required 1/2, and no amount of manipulation, for converting the distances to areas, can make it conform to the mirror boundary equation.

A second approach is to consider the distance from the edge as proportional to the characteristic crack dimension. This would mean that

the crack extension stage would predominate, and that the force would be proportional to the 3/2 power of the crack dimension (or distance from the edge), as expected for a half-penny crack[34,35]. The data does appear to cluster around a 3/2 slope, although it is impossible to know if the scatter in slopes is due to test or material variability, or is a natural material function. If the distance from the edge is truly an indicator of the force necessary to extend a crack, then this method may eventually make it possible to determine such factors as the fracture surface energy. IITRI determined K_{I_C} by the double torsion method for the materials tested here. See Table 4. Those values correlate to the ranking determined here, if the small differences between batches 3 and 1 are ignored.

If we then review all of the data, the strengths of these materials correlate with the densities, which correlate with the hardness, and the modulus correlates with the K_{I_C} which correlates with the distance from the edge. Also, a plot of the distance from the edge vs load gives roughly the slope 3/2 for all batches, the relationship between the force and extension of a half-penny crack. Intriguing as these results are, it would be overoptimistic at this time to suggest that anything other than a method for ranking materials or batches, by modulus or K_{I_C}, has been developed. An extremely limited range of materials was tested, and as the difficulties with the NC-132 material showed, some materials may be difficult to accomodate to this test. It is hoped that these results will encourage further exploitation of the hardness test.

ACKNOWLEDGMENT

We are indebted to Dr. R. Ruh of the Air Force Materials Laboratory and Dr. G. C. Walther of the IIT Research Institute for supplying the test materials and to Dr. H. P. Leighly of the University Missouri-Rolla for allowing the use of the Hardness tester.

Table 4

Material	Batch	K
NC-350	1	1.88 MN
	2	2.15
	3	2.08
NC-435	1	3.93
	3	3.91
	4	3.40

REFERENCES

1) H. Hertz, "Hertz's Miscellaneous Papers", MacMillan, London 1896.

2) J. Boussinesq, Discussed in "Theory of Elasticity", by Timoshenko and Goodier, McGraw-Hill, 1970. p398.

3) B. Lawn and R. Wilshaw, J. Mat. Sci., 10, 1049(1975). This is an extensive review paper.

4) B. Lawn and M. Swain, J. Mat. Sci., 10, 113(1975).

5) B. Lawn, S. M. Wiederhorn, and H. H. Johnson, J. Amer. Ceram. Soc., 58(9-10), 428(1975).

6) B. Lawn, E. R. Fuller, and S. M. Wiederhorn, J. Amer. Ceram. Soc., 59(5-6), 193(1976).

7) J. R. Varner and H. J. Oel, Glass. Tech. Ber., 48, 73(1975).

8) D. A. Krohn and D. P. H. Hasselman, J. Amer. Ceram. Soc., 54, 54(1971).

9) J. J. Melcholsky, S. W. Freiman, and R. W. Rice, J. Mater. Sci., 11, 1310(1976).

10) J. J. Melcholsky, R. W. Rice, and S. W. Freiman, J. Amer. Ceram. Soc., 57, 440(1974).

11) N. Shinkai, Jap. J. Apl. Phys. 14, 147(1975).

12) J. Congleton and N. J. Petch, Phil. Mag., 16,749(1967).

13) K. R. McKinney, J. Amer. Ceram. Soc., 56, 115(1973).

14) A. I. A. Abdel-Latif, R. C. Bradt, and R. E. Tressler, Int. J. Of Fract. 13(3), 349(1977).

15) L. R. F. Rose, Int. J. of Fract. 12(6), 799(1976). This is a review paper.

16) F. C. Frank and B. R. Lawn, Proc. Roy. Soc. Lond., A299, 291(1967).

17) B. R. Lawn, J. Appl. Phys., 39, 4828(1968).

18) A. G. Mikosza and B. R. Lawn, J. Appl. Phys., 42, 5540(1971).

19) T. R. Wilshaw, J. Phys. D:Appl. Phys., 4, 1567(1971).

20) J. S. Williams, B. R. Lawn, and M. V. Swain, Phys. Stat. Soli(a), 2

7(1979).

21) N. E. W. Hartley and T. R. Wilshaw, J. Mater. Sci., 8, 265(1973).

22) K. Phillips, Ph.D. Thesis, University of Sussex, 1975.

23) D. B. Marshall and B. R. Lawn, J. Amer. Ceram. Soc., 60(1-2), 86(1977).

24) A. G. Evans and E. A. Charles, J. Amer. Ceram. Soc., 59(7-8), 371(1976).

25) R. W. Rice, "The Science of Hardness Testing and Its Research Applications", ASM, 1973.

26) S. Palmqvist, Jernkontorets Ann., 141, 300(1957).

27) J. H. Westbrook, J. Amer. Ceram. Soc., 41, 433(1958).

28) H. E. Exner, Trans. Met. Soc. AIME, 245, 677(1969).

29) V. M. Glazov and V. N. Vigdorovich, "Microhardness of Metals and Semiconductors", Consultants Bureau, 1979.

30) E. K. Pavelcheck and R. H. Doremus, J. Mater. Sci., 9, 1802(1974).

31) Idem, J. Appl. Phys., 46, 4096(1975).

32) R. H. Doremus and W. A. Johnson, J. Mater. Sci., 13, 855(1978).

33) D. C. Larsen, "Property Screening and Evaluation of Ceramic Vane Materials", Interim Technical Report 3. Contract No. F33615-75-C-5196.

34) D. C. Larsen and G. C. Walther, "Property Screening and Evaluation of Ceramic Vane Materials", Interim Report 4.

35) D. C. Larsen and G. C. Walther, "Property Screening and Evaluation of Ceramic Turbine Engine Materials", Interim Technical Report 5.

36) B. R. Lawn and E. R. Fuller, J. Mater. Sci., 19, 2016(1975).

37) B. R. Lawn, T. Jesen, A. Arora, J. Mater. Sci., 11, 573(1966).

APPENDIX 1

<u>600 Grams</u>

Microns for Edge

		AV			AV
41T1A	40		31T1A	80	
	39			73	
	35			79	
	35			59	
	33	39		76	72
	37			75	
	32			78	
	42			60	
	45			67	
	47			62	
44T1A	45		33T1A	59	
	50			61	
	60			69	
	62			66	
	55	56		71	64
	47			62	
	65			58	
	57			70	
	63			60	
	51			68	
43T1A	47		32T1A	68	
	50			81	
	53			77	
	61			73	
	45	52		70	76
	55			82	
	49			79	
	57			72	
	48			80	
	60			78	

900 Grams

Microns for Edge

Sample	Microns	AV
41T1A	50	
	49	
	61	
	52	56
	49	
	52	
	60	
	63	
	65	
	59	
44T1A	81	
	72	
	71	74
	78	
	70	
	68	
	69	
	80	
	76	
	75	
43T1A	58	
	74	
	71	
	70	65
	63	
	66	
	57	
	61	
	63	
	68	

Sample	Microns	AV
33T1A	101	
	97	
	88	
	89	94
	91	
	98	
	95	
	90	
	91	
	100	
33T1A	91	
	86	
	79	83
	80	
	85	
	90	
	84	
	76	
	85	
	78	
32T1A	94	
	95	
	101	
	107	99
	104	
	110	
	92	
	96	
	89	
	97	

1,000 Grams

	Distance	AV
31T1A	101	
	104	
	98	
	89	96
	91	
	91	
	93	
	95	
	98	
	101	
33T1A	95	
	92	
	94	
	82	88
	91	
	84	
	86	
	93	
	81	
	83	
32T1A	98	
	99	
	97	
	109	105
	108	
	102	
	114	
	97	
	112	
	110	

<u>1,400 Grams Load</u>

	Microns for Edge	AV
41T1A	66	
	59	
	61	
	64	65
	70	
	68	
	62	
	71	
	67	
	61	
43T1A	87	
	85	
	89	
	80	82
	78	
	79	
	86	
	81	
	76	
	78	
44T1A	91	
	89	
	95	
	94	97
	103	
	102	
	98	
	96	
	100	
	102	

CERAMIC FRACTURE ANALYSIS
THROUGH BIAXIAL WEIBULL THEORY

by

William J. Craft

TABLE OF CONTENTS

APPENDIX

CERAMIC FRACTURE ANALYSIS THROUGH BIAXIAL WEIBULL THEORY

INTRODUCTION

The second project of this investigation was entitled "Strength characterization of brittle materials by means of simple-model destructive testing and by indentor-initiated crack growth".

A simple destructive test is appealing due to: (1) its decisiveness in quantifying failure; (2) its ability, if successful, in transferrability of results to complex designs and shapes; (3) its incorporation of size effects; and (4) in that it requires inexpensive testing equipment.

Literature on ceramics largely resides in three camps, fracture mechanics, continuum mechanics, and statistical failure. Fracture mechanics predictions of failure couple with elasticity theory where strain energy release rate is equated with a critical flaw length or size [1-2] frequently on the basis of standard 'crack' models such as interface delamination in the case composite material problems or as voids with sharp edges as in the case of the penny-shaped crack [2]. The difficulty in such approaches involving elasticity theory is the intractibility of the field equations even for linear elastostatic problems of simple geometry [3,4].

Finite element and difference methods offer some advances in the numerical techniques [5,6], in particular when coupled with ancillary methods that predict the stress intensity factor [7,8]. These methods all fail, however, to give an account of the customarily larger experimental scatter in apparent failure strengths of numerous grainy brittle materials.

The classical work on statistical failure by Weibull [9-11] has been applied to numerous stocastic processes capacitor discharge, fatigue time life, and

others largely due to the ease of use, simplicity of assumptions and non-stocastic processes as ceramic fracture in relation to a size effect. The major problems posed by this theory are the mathematical determination of the Weibull parameters, in particular for the three parameter distribution, and in the application of the results in a meaningful manner to the elasticity solution in order that a failure probability for the body can be found.

In order to develop a technique that would utilize Weibull Theory and an elasticity solution to generate fracture probabilities for a body of brittle material, it was necessary to investigate what techniques were proposed and used to generate the Weibull parameters to begin with.

Sample earlier work was soon found based on moment generating methods and on rank-order theory [12,13] to differ with results obtained by several iterative methods whereby the sume of squares of deviations were minimized (residuals) as is the case of standard least-squares techniques. Moment generation can involve large truncation error and rank-order assumptions should be significantly less precise than comparison with cumulative totals of experimental data. Further, although least-squares analyses tends to emphasize deviations in the experimental curve ordinates, incorporation of appropriate weights can counter this effect while preserving the methods basic simplicity.

Another problem with earlier techniques is that an assumption was generally required for multiaxial stress states necessitated by the uniaxial stress state under which data had been taken. Many persons assumed statistical independence in the actions of principal direction stresses, i.e., the probability of failure of an element equal to the product of probabilities of failure due to each principal direction stress [12]. Some writers have proposed

extensions of the uniaxial Weibul Theory based on fracture mechanics where distributions of penny-shaped cracks initiate failure [14,15,16].

In this report it was proposed to couple the results of multiaxial tests, notably biaxial ones, with an analysis and development of Weibull parameters obtained at specific stress ratios. This would aid the utilization of experimental data without the need to develop further fracture mechanics assumptions. Further, it has been hoped that such a technique when coupled with effective testing of standard specimens and stress fields could help lend a standardization in the analysis of experimentally obtained statistical fracture.

DEVELOPMENT OF SIMPLE TEST ANALYSES

An extension to publications where standard specimen shapes are used to generate Weibull statistics [17,18] are developed below where the three parameter Weibull fit is used.

UNIAXIAL CONSTANT STRESS FIELDS

The Weibull distribution for a unaxial constant stress state is given below:

$$F(\sigma) = 1 - e\,x\,p\left(-\kappa \int \left(\frac{\sigma\ \sigma_u}{\sigma_\theta}\right)^m \frac{dU}{U_{un}}\right)$$

U_{un} = Unit Volume

κ, σ_u, σ_θ, m = Weibull Parameters which can be expressed uniquely as three independent ones.

For unaxial constant stress, is a principal stress, independent of the volume so that

$$F(\sigma) = 1 - \exp\left(-\frac{V_q}{V_{un}}\left(\frac{\sigma-\sigma_u}{\sigma_\theta}\right)^m\right)$$

Upon using $C = \dfrac{V_q}{V_{un}(\sigma_\theta)^m}$

$$F(\sigma) = 1 - \exp(-c(\sigma-\sigma_u)^m)$$

The simpliest least-squares solution is obtained by noting that taking logs of this equation twice gives:

$$\ln\ln\left(\frac{1}{1 - F(\sigma)}\right) = \ln(c) + m \ln(\sigma-\sigma_u)$$

This is a linear relationship allowing a linear least-square analysis to be invoked for

$$y = A+Bx$$

where $A = \ln(c)$

$B = m$

$x_\iota = x(\sigma_\iota) = \ln(\sigma_\iota-\sigma_u)$

$y(\sigma_\iota) = y_\iota = \ln\ln\left(\frac{1}{1 - F(\sigma_\iota)}\right)$

$\iota = 1,2,\ldots, n$ - the number of data points.

The solution that minimizes the residual is:

$$A = \left((\Sigma w_\iota y_\iota x_\iota)\ (\Sigma w_\iota) - (\Sigma w_\iota x_\iota)\ (\Sigma w_\iota y_\iota)\right)/D$$

$$B = \left((\Sigma w_\iota x_\iota^2)\ (\Sigma w_\iota y_\iota) - (\Sigma w_\iota x_\iota)\ (\Sigma w_\iota x_\iota y_\iota)\right)/D$$

$$D = (\Sigma w_\iota)\ (\Sigma w_\iota x_\iota^2) - (\Sigma w_\iota x_\iota)^2$$

Where all summations are $\iota = 1,\ldots, n$.

In case equal weights are desired, the quantity Σw_ι is replaced by n, otherwise, $w_\iota = 1$. The obvious difficulty with this analysis is that A and B are dependent on σ_u. To determine the best σ_u, an iterative programming scheme, Appendix I, was developed which recomputes the residual sum of squares and chooses the σ_u minimizing this function. By a slight modification this scheme can be used to redefine the weights iteratively to emulate any curve-fitting power law.

PURE BENDING STRESS FIELD IN A RECTANGULAR SECTION BAR

Tests utilizing bending stress distributions are common and generally less expensive than ceramic tension tests — particularly in the base of three and four point loading tests. Following the two parameter derivation [17], where the stress distribution is variable and must be integrated over the volume in bending, the probability of failure is given by:

$$P = 1 - \exp -\left(\frac{V}{2(m+1)}\left(1 - \frac{\sigma_u}{\sigma_b}\right)\left(\frac{\sigma_b - \sigma_u}{\sigma_0}\right)^m\right)$$

Where $V = bLh$

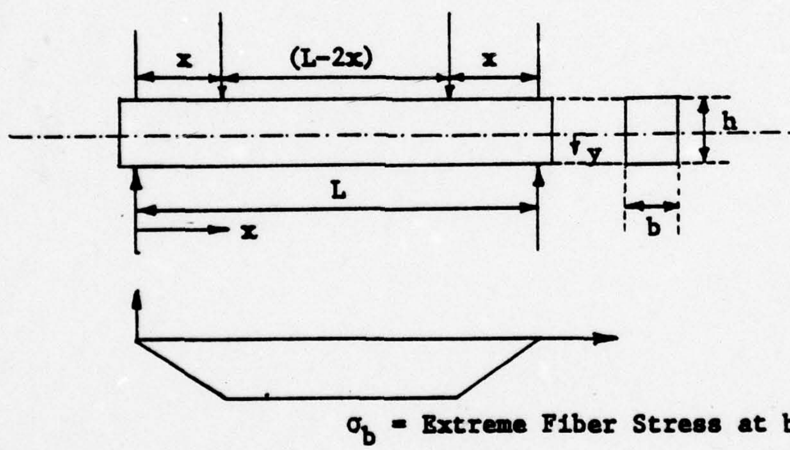


Figure 1. Prismatic beam under 4-point loading and distribution of extreme fiber stress

Fig. 1 depicts this case with its test volume. This distribution follows the same form of the case of a constant uniaxial stress field and may be solved interatively by the technique where upon taking $\ell n \ell n$, one has

$$\ell n \ell n\left(\frac{1}{1 - F(\sigma_l)}\right) = \ell n(V/2) - \ell n(m+1) + (m+1)\ell n(\sigma_{b_l} - \sigma_u) - \ell n(\sigma_{b_l}) - m\ell n(\sigma_0)$$

In this case

$$A = \ln(V/2) - \ln(m+1) - m \ln(\sigma_0)$$

$$B = m+1 .$$

PURE BENDING FOR A CIRCULAR CROSS-SECTION ROD

This case is related to the others except that the integration is conducted over a circular prison for the section in pure bending. The resulting integral form:

$$\int_{V_T} (my - \sigma_u I_y)^m \sqrt{R^2 - y^2}\ dy$$

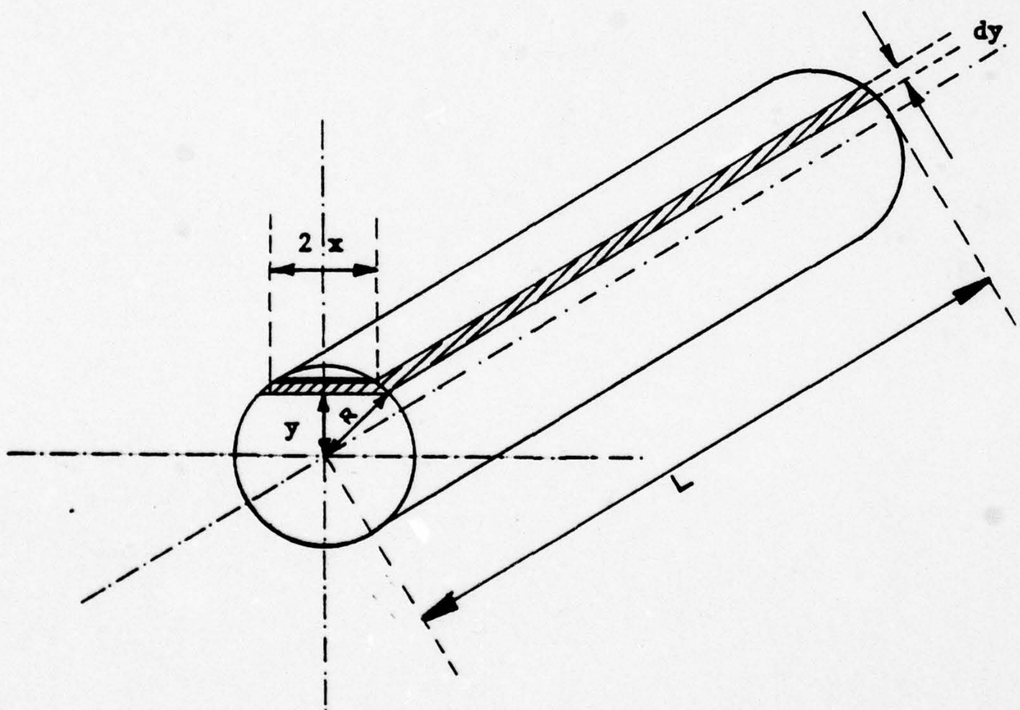


Figure 2. Bending for a circular cross-section rod

must be evaluated as a function of σ_u and M before the previously discussed **$\ln\ln$** least-square procedure can be applied. Fig. 2 depicts this case and its test volume, V_T.

PURE TORSION FOR A CIRCULAR BAR

Under the action of pure torsion, the stress field $\tau_{rz} = \frac{Tr}{J}$ results from which the principal direction stresses are:

$$\sigma_1 = \tau_{rz},\ \sigma_2 = -\tau_{rz}\ .$$

In order that a Weibull distribution based on this test can be evaluated, two parameters, discussed in the next section must be defined:

$$\sigma^2 = \sigma_1^2 + \sigma_2^2$$

and $\theta = \mathrm{Tan}^{-1}\ (\frac{\sigma_2}{\sigma_1}) = 135^\circ$

They are the intensity of the biaxial stress field, σ , and the angle on the biaxial stress distribution that the point on the fracture surface makes with the σ_1 axis, as depicted below:

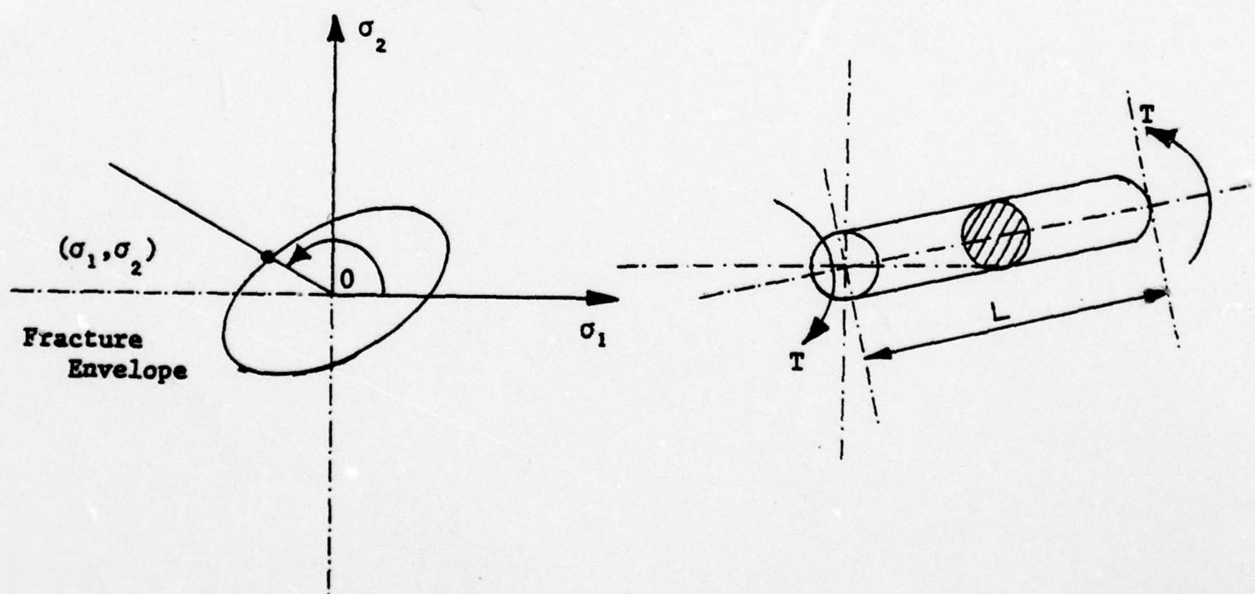


Fig. 3. Biaxial Stress Envelope

In this case, the stress distribution is integrated over the volume as before resulting in

$$P(\sigma) = 1 - e^{-B_n}$$

Where $$B_n = \frac{2\pi \ell}{(m+1)\sigma_0^m} \left\{ \frac{(\sigma - \sigma_n) r^{\frac{m+1}{2}}}{\sigma} - \frac{(\sigma - \sigma_u) r^{\frac{m+2}{2}}}{\sigma^2 (m+2)} \right\}_{\frac{l\sigma_u}{T}}^{R_0}$$

This result has the same form as the case of the uniaxial tension field and can be correlated with the three parameter Weibull distribution by the Appendix I program.

FOURIER ANALYSIS OF BIAXIAL DATA

All three Weibull parameters are functions of the principal direction stress ratio or θ where

$$\theta = \mathrm{Tan}^{-1} \left(\frac{\sigma_2}{\sigma_1}\right)$$

Thus each Weibull parameter c, σ_0, m could be written by

$$c = c(\theta)$$

$$\sigma_0 = \sigma_0(\theta) \qquad \text{or} \qquad We = \begin{Bmatrix} c \\ \sigma_0 \\ m \end{Bmatrix} = We(\theta)$$

$$m = (\theta) = We(\theta \pm 2n\pi)$$

invoking periodicity we also have

$$We(\theta) = We(\theta \pm 2n\pi)$$

This same theory can be developed for the solid angle representation in

the case of three principal direction stress components; that extension is based on orthogonal polynomials and representation completeness of harmonic analysis [3,4,19].

Once the three Weibull parameters are known for a variety of θ_ι , the equivalent parameters can be obtained for any angle by means of trigonometric interpolation. The computer program FOURIE, Appendix II, requires the Weibull set We_ι at a variety of θ_ι. At this point, the highest harmonic obtainable for the number of data points is used for the interpolation series or the highest harmonic is used which satisfies the null hypothesis condition [20]. The advantage of this program is that the coefficients to the Fourier harmonics, once generated, can be used for each element stress ratio without additional coefficient computation.

The computer program would naturally be much simplier with equally-spaced θ_ι data points but such is not sufficiently general, hence orthogonality cannot be invoked without the creation of special orthogonal sequences.

Basically given n (n odd) angles θ_ι $\iota=1,\ldots,n$, a set of functions can be generated with $L=\frac{n-1}{2}$ to define $y_m(\theta)$ where M L. In this case:

$$y_m(\theta) = \frac{a_0^m}{2} + \sum_{j=1}^{m} a_j^m, \cos(j\theta) + b_j^m \sin(j\theta) \ .$$

The residual function is:

$$H(a_0^m,\ldots,a_m^m) = \sum_{\iota=1}^{n} w(\theta_\iota)\left\{\bar{f}_\iota - y_m(\theta)\right\}^2$$

This function is dependent on the weights $w(\theta_\iota)$ associated with each Weibull 3-set $We(\theta_\iota)=\bar{f}_\iota$. It can be shown that evaluation of the above, minimizing H, leads to the partitioned matrix equation:

$$\begin{bmatrix} \frac{1}{2}\Sigma w_\iota & \Sigma w_\iota \cos(\theta_\iota) \ldots. & \Sigma w_\iota \sin(\theta_\iota) \ldots. \\ \frac{1}{2}\Sigma w_\iota \cos(\theta_\iota) & \Sigma w_\iota \cos^2(\theta_\iota) \ldots. & \Sigma w_\iota \cos(\theta_\iota)\sin(\theta_\iota) \\ \vdots & \vdots & \vdots \\ \frac{1}{2}\Sigma w_\iota \sin(\theta_\iota) & \Sigma w_\iota \cos(\theta_\iota)\sin(\theta_\iota) \ldots. & \Sigma w_\iota \sin^2(\theta_\iota) \ldots. \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ b_1 \\ \vdots \\ b_n \end{bmatrix} \begin{bmatrix} \Sigma w_\iota \bar{f}_\iota \\ \Sigma w_\iota f_\iota \cos(\theta_\iota) \\ \vdots \\ \Sigma w_\iota f_\iota \sin(\theta_\iota) \\ \vdots \end{bmatrix}$$

Summations are $\iota = 1,\ldots,n$.

For brevity only the first term in each set is listed here. Modification of first row terms leads to symmetry of the coefficient matrix.

FINITE ELEMENT ANALYSIS

The remaining task in the use of biaxial Weibull statistics is the use of a general stress analysis method to find the stress distribution in a body so that this theory can be applied. The crux of this process is a finite element analysis model written by J. Brisbane [21] for bodies of revolution loaded axisymmetrically. This code has been modified to create files on the principal stresses for each element as well as volume for a particular material designation. The modified program, Appendix III, does generate triaxial stress data but for most loading conditions, one stress component, in or near the r-coordinate direction can usually be neglected.

As with all programs of this type, input information is comprehensive and complex. Only changes are described when they relate to the Weibull analysis. One additional parameter, the material number N is required, this is the number of the material subject to the Weibull modeling.

In order to analyse Weibull statistics, two additional disk files have been added, unit 43 and unit 47. Unit 43 stores the location of each model point of each element of material N. It also stores the volume on each record. Unit 47 records for each material N element, the principal stresses. These files are saved after completion of execution so that a Weibull summation can be performed in conjunction with the output of Appendix II. The program of Appendix requires data from the Appendix I program or a modification thereof.

CONCLUSION

During this next years program, this Weibull analysis will be performed and compared to data derived from experiments on test specimens. Each experiment is designed to provide Weibull parpameters for specific principal stress-ratios and will complement entirely the present work. In order to compare this biaxial theory to other methods, one test specimen is being designed with a variety of stress ratios present within its volume. Its fracture behavior will be noted and compared to predictions based on this and the independence of principal direction theory of failure. Thus what has remained is an experimental confirmation of this theory which should be provided within the 1978 - 1979 period.

REFERENCES

[1] Erdogan, F. and Ozbek, T., "Stresses in Fiber-Reinforced Composites with Imperfect Bonding," Journal of Applied Mechanics, Vol. 35, No. 4, 1969, p. 865.

[2] Sueddon, I. N., Fourier Transforms, McGraw-Hill Book Co., Inc., New York, N. Y., 1951.

[3] Blenkarn, Kenneth A., "Fourier Integral Solutions to Radially Symmetric Elasticity Problems," Ph.D. Dissertation in Mechanical Engineering, The Rice Institute, Houston, Texas, May, 1960.

[4] Lure, A. I., Three-Dimensional Problems of the Theory of Elasticity, Interscience Publishers, a division of John Wiley & Sons, Inc., 1964.

[5] Zienkiewicz, O. C., The Finite Element Method in Engineering Science, McGraw-Hill, London, 1971.

[6] Martin, Harold C. and Carey, Graham F., Introduction to Finite Element Analysis, McGraw-Hill Book Co., Inc., 1973.

[7] Becker, E. B., and Dunham, R. S., "Some Stress Intensity Calculations Using Finite Elements," Finite Element Methods in Engineering, The University of New South Wales, Sydney, Australia, 1974.

[8] Stern, Morris and Soni, Mahan Lal, "on the Computation of Stress Intensity Factors in Fibre Composite Media Using a Contour Integral Method, "EMRL No. 1110, Department of Aerospace Engineering and Engineering Mechanics," University of Texas at Austin, Austin, Texas, August, 1974.

[9] W. Weibull, "A Survey of Statistical Effects in the Field of Material Failure", Applied Mechanics Reviews, Vol. 5 #11, November, 1952.

[10] W. Weibull, "A Statistical Distribution Function of Wide Applicability," Journal of Applied Mechanics, September, 1951.

[11] "Weibull's Statistical Theory of Strength, Appendix III," WADC T-R 53-50 Pt 1.

[12] "The Analysis of Brittle Components," Appendix I to Chapter 5, Report of the Materials Advisory Board Ad Hoc Committee on Ceramic Processing, Washington, D.C.

[13] Gregory and Spruill, "Structural Reliability of Re-Entry Vehicles Using Brittle Materials in the Primary Structure," IAS Aerospace Systems Reliability Symposium, 1962.

[14] S. B. Batdorf, "A Statistical Theory for Failure of Brittle Materials

Under Combined Stresses", AIAA Paper, #73-381.

[15] S. B. Batdorf, "Weibull Statistics for Polyaxial Stress State," Journal of American Soc., January, 1974.

[16] A. G. Evans, "A General Approach for the Statistical Analysis of Multiaxial Fracture," August, 1978.

[17] N. A. Weil, I. M. Daniel, "Analysis of Fracture Probabilities in Nonuniformly Stressed Brittle Materials," Journal of American Soc., Vol. 47, #6, June, 1964.

[18] Daniel, I. M. and Weil, N. A., "The Influence of Stress Gradient Upon Fracture of Brittle Materials," Journal of American Society of Mechanical Engineers, Paper No. 63-WA-228.

[19] Rubinstein, Zolman, A Course in Ordinary and Partial Differential Differential Equations, Academic Press, 1969.

[20] Ralston, Anthony, A First Course in Numerical Analysis, McGraw-Hill Book Co., Inc., 1965.

[21] Brisbane, John J., "Heat Conduction and Stress Analysis of Solid Propellant Rocket Motor Nozzles," Technical Report S-198, Rohm and Haas Co., Redstone Research Laboratory, Huntsville, Ala., 1969.

```
C PROGRAM WEIBUL THREE PARAMETERS
C THE PARAMETERS ARE SIGMA0, M ,AND C
C THIS PROGRAM COMPUTES ALL PARAMETERS BY
C LINEAR REGRESSION METHOD. FOR NOTATION SIGMA0=SIGO
        DIMENSION X(20),F(20),W(20),EK(20),P(20),WN(20),WM(20)
        WRITE(5,10)
10      FORMAT(1H,8HGO AHEAD)
C GO AHEAD REQUIRES YOU TO TYPE NO. OF DATA FILE
        READ(5,*) ND
        READ(ND,*) N
C ND=NUMBER OF DATA FILE
C N=NUMBER OF WEIBUL POINT PAIRS
C W=ARBITRARY WEIGHTS
C X=FRACTURE STRESS VALUES
        READ(ND,*)  ((X(I),F(I),W(I)),I=1,N)
        DO 1 I=1,N
1       WRITE(5,20) (I,X(I),F(I),W(I))
20      FORMAT(1X,°X(I),F(I),W(I) FORI=°,I2,3(1X,E12.6))
        DO 2 I=1,N
        EK(I)=ALOG(ALOG(1./(1.-F(I))))
2       CONTINUE
        NN=5
        JK=0
        INT=10
        RESO=1.E+30
        BEG=-.99*X(1)
        EMD=.99*X(1)
98      H=(EMD-BEG)/FLOAT(INT)
        DO 95 I=0,INT
        SIK=BEG+FLOAT(I)*H
        DO 96 II=1,N
96      P(II)=ALOG(X(II)-SIK)
        CALL LIN(W,WN,P,EK,N,SIK,A,B,RES)
        IF(RES.GT.RESO) GO TO 95
        RESO=RES
        SIG=SIK
        AG=A
        BG=B
        J=I
95      CONTINUE
        WRITE(5,80) AG,BG,RESO,SIG,SIGO
C REDIFINE EMD AND BEG AND CONTINUE
        IF (JK.GT.NN) GO TO 97
        JK=JK+1
        IF (J.EQ.0) J=1
        IF (J.EQ.INT) J=INT-1
        EMD=BEG+FLOAT(J+1)*H
        BEG=EMD-2.*H
        RESO=RES
        SIGO=SIG
        GO TO 98
80      FORMAT(2X,°AG=°,E12.6,7X,°BG=°,E12.6/,2X,°RESO=°,E12.6,
```

```
     1 5X,°SIG=°,E12.6,5X,°SIGO=°,E12.6)
97    EC=EXP(AG)
      ESIGO=SIG
      EM=BG
      WRITE(5,100)EC,EM,SIG
100   FORMAT(/,2X,°EC=°,E12.6,5X,°EM=°,E12.6,5X,°SIG=°,E12.6/)
C CALCULATED PROBABILITIES ARE LISTED AS BELOW
      DO 130 II=1,N
130   P(II)=ALOG(X(II)-SIG)
      CALL LIN(W,WM,P,EK,N,SIG,A,B,RES)
      DO 108 I=1,N
      PCAL=1.0-EXP(-EC*(X(I)-ESIGO)**EM)
108   WRITE(5,110) PCAL,F(I),W(I),WM(I)
110   FORMAT(1X,°PCAL,F(I),W(I),WM(I)=°,E12.6,3(1X,E12.6))
      STOP
      END
      SUBROUTINE LIN(W,WN,P,EK,N,SIG,A,B,RES)
      DIMENSION  W(20),P(20),EK(20),WN(20)
      C=0.
      D=0.
      E=0.
      G=0.
      H=0.
      DO 4 I=1,N
      C=C+(W(I))
      D=D+(W(I)*P(I))
      E=E+(W(I)*P(I)*P(I))
      G=G+(W(I)*EK(I))
      H=H+(W(I)*EK(I)*P(I))
4     CONTINUE
      DEN=E*C-D*D
      A=((E*G-D*H)/(DEN))
      B=((C*H-D*G)/(DEN))
C THE CONSTANTS A AND B ARE KNOWN
C NOW FIND RESIDUE
      SUMM=0.
      DO 5 I=1,N
      WN(I)=(A+B*P(I))-EK(I)
5     SUMM=SUMM+W(I)*WN(I)*WN(I)
      RES=SUMM
      RETURN
      END
```

```
C THIS IS A BIAXIAL PROGRAM
        IMPLICIT DOUBLE PRECISION(A-H,O-Z)
        DIMENSION TH(20),EM(20),W(20),A(20,20),B(20),XX(20)
     1  ,XK(20)
C FOURIER LEAST SQUARE APPROXIMATION PROGRAM
C ND=THE FILE NAME AS IN ND=5(THIS TERMINAL) OR
C FORND.DAT=THE DISK FILE
        WRITE(5,40)
40      FORMAT(2X,'ENTER DATA FILE NO.')
        READ(5,*) ND
        WRITE(5,70)
70      FORMAT(1X,'NOW ENTER NO. OF POINTS ')
        READ(5,*) N
C N=NUMBER OF WEIBULL POINT PAIRS OR STRESS/FRACTURE/
C PERCENT PAIRS.
        READ(ND,*) ((TH(I),EM(I),W(I)),I=1,N)
        DO 50 I=1,N
50      WRITE(5,60) (I,TH(I),EM(I),W(I))
60      FORMAT(1X,'TH(I),EM(I),W(I) FORI=',I2,3(1X,E13.6))
C W=THE WEIGHT AT EACH THETA
C THETA IS IN RADIANS
C EM=THE WEIBUL MODULUS
C  TO MAKE THE MATRIX SYMMETRIC WE MULTIPLY TOP ROW
C BY .5 AFTER FORMULATION OF MATRIX
C DATAS ARE UNEQUALLY SPACED
C FOURIER APPROXIMATION FITTING THE DATA POINTS.
C THEN THE NORMAL EQUATIONS ARE SET UP.
C THEN THEY ARE SOLVED AND TESTED FOR SIGNIFICANCE
C AND THE FOURIER APPROXIMATION IS INCREASED BY ONE DEGREE IS
C NEEDED AND THE PROCESS REPEATED IF NEEDED.
C ITER=THE NUMBER OF TIMES THE PROCESS IS EXECUTED.
        NP=5
        MP=N/2
        MP=2*MP
        ITER=0
        SIGO=30E+30
        M=1
C       SET UP THE FIRST MATRIX
13      I=1
        B(I)=PHO(I,N,W,EM,TH)
        DO 51 I=2,M+1
        K=I-1
51      B(I)=RHO(K,N,W,EM,TH)
        DO 52 I=M+2,2*M+1
        K=I-M-1
52      B(I)=GHO(K,N,W,EM,TH)
        WRITE(5,204)
204     FORMAT(/,10X,'RIGHT HAND SIDE MATRIX')
        WRITE(5,205)  (I,B(I),I=1,2*M+1)
205     FORMAT(/,10X,I2,4X,E13.6)
        I=1
        J=1
```

```
          A(I,J)=RMC(I,J,N,W,EM,TH)
          I=1
          DO 53 J=1,M
          K0=I
          L0=J+1
53        A(K0,L0)=AMC(J,K0,L0,N,W,EM,TH)
          I=1
          DO 54 J=1,M
          K4=I
          L4=J+1+M
54        A(K4,L4)=BMC(J,K4,L4,N,W,EM,TH)
          DO 55 I=1,M
          DO 55 J=1,M
          K1=I+1
          L1=J+1
55        A(K1,L1)=DMC(I,J,K1,L1,N,W,EM,TH)
          DO 56 I=1,M
          DO 56 J=1,M
          K2=I+1
          L2=J+1+M
56        A(K2,L2)=GMC(I,J,K2,L2,N,W,EM,TH)
          DO 57 I=1,M
          DO 57 J=1,M
          K3=I+1+M
          L3=J+1+M
57        A(K3,L3)=HMC(I,J,K3,L3,N,W,EM,TH)
          DO 58 I=1,2*M
          DO 58 J=I+1,2*M+1
58        A(J,I)=A(I,J)
          A(I,J)=A(J,I)
          WRITE(5,219)
219       FORMAT(/,6X,°MATRIX OF COEFICIENTS°/,7X,°I°,5X,°J°,
     1    9X,°A(I,J)°,/)
          WRITE(5,220) ((I,J,A(I,J),J=1,2*M+1),I=1,2*M+1)
220       FORMAT(6X,I2,4X,I2,5X,E13.6)
C NOW SOLVE THE NORMAL MATRIX EQUATIONS
C NP IS THE NUMBER OF PRINTER
          M1=2*M+1
          CALL SOLVE(A,XX,B,M1,DET,NP)
          WRITE(5,225) M,(XX(I),I=1,M1)
225       FORMAT(2X,°M=°,I5,/,1X,°XX(I)=°,4(3E13.6))
C NOW TEST THE STATISTICAL SIGNIFICANCE OF INCREASING M
          DO 59 I=1,2*M+1
59        XK(I)=XX(I)
          IF(MP.EQ.N) GO TO 101
          IF(MP.LT.N) GO TO 102
101       IF(2*M+1.EQ.(MP-1)) GO TO 99
102       IF(2*M+1.EQ.N) GO TO 99
C COMPUTE DELTA**2 THEN SIGMA**2
          DELN=0.
          DO 61 I=1,N
          TEMP=XX(1)/2
```

```
            DO 62 J=1,M
            J1=J+1
            J2=M+1+J
62          TEMP=TEMP+(XX(J1)*COS(J*TH(I))+XX(J2)*SIN(J*TH(I)))
61          DELN=DELN+W(I)*(EM(I)-TEMP)**2
C NOW COMPUTE SIGMA**2
            XT=N-2*M-1
            SIGN=DELN/XT
            WRITE(5,200) DELN,SIGN,SIGO
200         FORMAT(2X,°DELN=DELTA**2=°,E13.6/,
         1  2X,°SIGN=SIGMA**2=°,E13.6,2X,°SIGO=°,E13.6)
            IF(SIGN-SIGO) 11,99,99
11          SIGO=SIGN
            M=M+1
            GO TO 13
99          STOP
            END
C FUNCTION PHO COMPUTES THE ELEMENTS IN R.H.S. MATRIX
            FUNCTION PHO(I,N,W,EM,TH)
            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
            DIMENSION W(20),EM(20),TH(20)
            SUMM=0.
            DO 1 L=1,N
1           SUMM=SUMM+(W(L)*EM(L))*.5
            PHO=SUMM
            RETURN
            END
C FUNCTION RHO COMPUTES THE ELEMENTS IN R.H.S. MATRIX
            FUNCTION RHO(J,N,W,EM,TH)
            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
            DIMENSION W(20),EM(20),TH(20)
            SUMM=0.
            DO 1 L=1,N
1           SUMM=SUMM+(W(L)*EM(L)*COS(J*TH(L)))
            RHO=SUMM
            RETURN
            END
C FUNCTION GHO COMPUTES THE ELEMENTS IN R.H.S. MATRIX
            FUNCTION GHO(J,N,W,EM,TH)
            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
            DIMENSION W(20),EM(20),TH(20)
            SUMM=0.
            DO 1 L=1,N
1           SUMM=SUMM+(W(L)*EM(L)*SIN(J*TH(L)))
            GHO=SUMM
            RETURN
            END
C ALL THE FUNCTIONS BELOW  FORMS THE L.H.S. MATRIX
            FUNCTION RMC(I,J,N,W,EM,TH)
            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
            DIMENSION W(20),EM(20),TH(20)
            SUMM=0.
```

```
      DO 1 L=1,N
1     SUMM=SUMM+(W(L))*.25
      RMC=SUMM
      RETURN
      END
      FUNCTION AMC(J,K0,L0,N,W,EM,TH)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION W(20),EM(20),TH(20)
      E=0.
      DO 1 L=1,N
    1 E=E+(W(L)*COS(J*TH(L)))*.5
      AMC=E
      RETURN
      END
      FUNCTION BMC(J,K4,L4,N,W,EM,TH)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION W(20),EM(20),TH(20)
      SUMM=0.
      DO 1 L=1,N
1     SUMM=SUMM+(W(L)*SIN(J*TH(L)))*.5
      BMC=SUMM
      RETURN
      END
      FUNCTION DMC(I,J,K1,L1,N,W,EM,TH)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION W(20),EM(20),TH(20)
      SUMM=0.
      DO 1 L=1,N
1     SUMM=SUMM+(W(L)*COS(J*TH(L))*COS(I*TH(L)))
      DMC=SUMM
      RETURN
      END
      FUNCTION GMC(I,J,K2,L2,N,W,EM,TH)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION W(20),EM(20),TH(20)
      SUMM=0.
      DO 1 L=1,N

1     SUMM=SUMM+(W(L)*SIN(J*TH(L))*COS(I*TH(L)))
      GMC=SUMM
      RETURN
      END
      FUNCTION HMC(I,J,K3,L3,N,W,EM,TH)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION W(20),EM(20),TH(20)
      SUMM=0.
      DO 1 L=1,N
1     SUMM=SUMM+(W(L)*SIN(J*TH(L))*SIN(I*TH(L)))
      HMC=SUMM
      RETURN
      END
```

```
C      PROGRAM 00046(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE10,TAPE11,
C     1TAPE12,TAPE13,TAPE14,TAPE15,TAPE16,PUNCH)
       IMPLICIT REAL*8(A-H,O-Z)
       REAL*8 TITLE,WORD1,WORD2,WORD3
       REAL NU21,NU31,NU32
C**AMG054 FINITE ELEMENT STRESS ANALYSIS OF ROCKET NOZZLES
       INTEGER CODE,ERR,PCODE,BW,TOPT
       DIMENSION R(30,50),Z(30,50),CODE(30,50),TITLE(13),UF(100),WF(100)
     1  ,TANF(100),IMAX(50),IMIN(50),BC(100,3),CONPR(16,15),IP(200)
     2  ,JP(200),P(200),TAU(200),PCODE(200),NEQ(50),            S(8,8)
     3  ,F(8),C(4,4),A(120,80),B(120),U(30,50),W(30,50)
     4  ,RT(2000),ZT(2000),TEMP(2000),PSI(4),SSMAX(14),SSMIN(14),IJSS(14
     5  ,4)
       COMMON/TEM/A
       COMMON UF,WF,TANF,JMIN,JMAX,ERR,MAX,R,Z,CODE,TITLE,IMAX,IMIN,
     1       CONPR,IP,JP,P,TAU,PCODE,BW,NEQ,JRAN,       S,F,NPCARD
     2  ,TOPT,NP,MN
       COMMON/VARPRO/TABLE(12,10,5)
       EQUIVALENCE  (BC,UF),(A(1),U),(A(1501),W),(A(3001),RT)
     1  ,(A(5001),ZT),(A(7001),TEMP)
       DATA WORD1/6HH ONLY/,WORD2/6HEND OF/,WORD3/6HAMG054/
       CALL ERRSET(209,256,-1,1,0)
       KPLOT=0
       MAX = 30
       JR1 = 120
         READ(40,101) N1,N2,N3
C N1=INPUT DATA FILE NO.=40
C N2=OUTPUT DATA FILE NO.=41
C N3=MAT NO FOR WHICH WEIBULL STS. IS TO BE DONE!FOR WEIBUL
     4 READ(40,100)(TITLE(I),I=1,13)
       IF(TITLE(13).NE.WORD1)WRITE(46)TITLE
  1000 READ(40,101) JMIN,JMAX,       NCONT,TOPT,NTABLE
       REWIND 42
       REWIND 44
       REWIND 45
         CALL ERRSET(1)
       WRITE(41,205) (TITLE(I),I=1,13)
       WRITE(41,206) JMIN,JMAX,       NCONT,NTABLE
       ITOPT=TOPT+1
       GO TO(3000,3001,3002,3003,3004,3000),ITOPT
 3001  WRITE(41,218)
       GO TO 3000
 3002  WRITE(41,219)
       GO TO 3000
 3003  WRITE(41,220)
       GO TO 3000
 3004  WRITE(41,221)
 3000  CONTINUE
       JRAN=JMAX-JMIN+1
C*****  INITIALIZE
       DO 2 J=1,JRAN
       DO 1 I=1,MAX
```

AD-A062 315 NORTH CAROLINA AGRICULTURAL AND TECHNICAL STATE UNIV --ETC F/G 9/5
A STUDY OF TWO-DIMENSIONAL RECURSIVE DIGITAL FILTERS AND A STUD--ETC(U)
NOV 78 N00014-77-C-0199
UNCLASSIFIED NL

2 OF 2
ADA
062315

END
DATE
FILMED
3 -79
DDC

```
      R(I,J)=0.
      Z(I,J)=0.
    1 CODE(I,J)=0
      IMIN(J)= 1000
    2 IMAX(J)= 0
      DO 3 I=1,100
      DO 3 J=1,3
    3 BC(I,J)=0.
      ERR=0
C***    READ NODAL POINT DATA
      CALL MESH
C     CALL GRIDSC(R,Z,CODE,IMIN,IMAX,JRAN,KPLOT,TITLE)
      IF(TOPT.EQ.5) TOPT=0
      IF(ERR.EQ.0) GO TO 6
      WRITE(41,212) ERR
    5 READ(40,100) (TITLE(I),I=1,13)
      IF(WORD2.NE.TITLE(1).AND.WORD3.NE.TITLE(1)) GO TO 5
      GO TO 2001
    6 IF (TITLE(13) .EQ. WORD1) GO TO 5
C***     READ PRESSURE CARDS
      CALL PRESBC(IP,JP,P,TAU,PCODE,NPCARD,ERR)
      IF(ERR.NE.0)GO TO 5
      IF(NPCARD.EQ.0) GO TO 8
      WRITE(41,207)
C***    ORDER THE PRESSURE CARDS BY INCREASING I AND J
      IF(NPCARD.EQ.1) GO TO 706
      N1=NPCARD-1
      DO 704 N=1,N1
      IS= IP(N)
      JS=JP(N)
      NS=N
      M1= N+1
      DO 703 M=M1,NPCARD
      IF(JP(M)-JS)702,701,703
  701 IF(IP(M).GT.IS)GO TO 703
  702 IS=IP(M)
      JS=JP(M)
      NS=M
  703 CONTINUE
      TS=TAU(NS)
      PS=P(NS)
      IPCS=PCODE(NS)
      IP(NS)=IP(N)
      JP(NS)=JP(N)
      P(NS)=P(N)
      TAU(NS)=TAU(N)
      PCODE(NS)=PCODE(N)
      IP(N)=IS
      JP(N)=JS
      P(N)=PS
      TAU(N)=TS
      PCODE(N)=IPCS
```

```
  704 WRITE(41,208)IS,JS,PS,TS,IPCS
  706 WRITE(41,208)IP(NPCARD),JP(NPCARD),P(NPCARD),TAU(NPCARD),PCODE(NPCA
     1  RD)
C***     READ CONTINUUM MATERIAL PROPERTIES
    8 IF(NCONT.EQ.0) GO TO 13
      WRITE(41,203)
      DO 9 N=1,NCONT
      READ(40,103) MN,E1,E2,E3,NU21,NU31,NU32,PHI,G13,ALPHA1,
     1   ALPHA2,ALPHA3,CONPR(15,MN),CONPR(16,MN)
      WRITE(41,204) MN,E1,NU21,ALPHA1,G13,PHI,E2,NU31,ALPHA2,
     1   CONPR(15,MN),E3,NU32,ALPHA3,CONPR(16,MN)
      CALL PROP(E1,E2,E3,NU21,NU31,NU32,G13,PHI,ALPHA1,
     1  ALPHA2,ALPHA3,C,PSI)
      ICOUNT=0
      DO 801 II=1,4
      CONPR(II+10,MN)=PSI(II)
      DO 801 JJ=II,4
      ICOUNT=ICOUNT+1
  801 CONPR(ICOUNT,MN)=C(II,JJ)
    9 CONTINUE
   13 IF(NTABLE.EQ.0)GO TO 11
C***  READ MATERIAL PROPERTY VS. TEMPERATURE TABLES
      WRITE(41,215)
      DO 10 N=1,NTABLE
      READ(40,104) MTN,NTEMP,BFR,BFZ,PHI,TO
      WRITE(41,216) MTN,BFR,BFZ,PHI
      MTN = MTN - 15
      DO 1001 N1=1,NTEMP
      READ(40,105)(TABLE(II,N1,MTN),II=1,11)
 1001 WRITE(41,217) (TABLE(II,N1,MTN),II=1,11)
      TABLE(12,1,MTN)=NTEMP
      TABLE(12,2,MTN)=BFR
      TABLE(12,3,MTN)=BFZ
      TABLE(12,4,MTN)=MTN+ 15
      TABLE(12,5,MTN)=TO
      TABLE(12,6,MTN) = PHI
   10 CONTINUE
C***  DETERMINE TEMPERATURES FOR CONTINUUM ELEMENTS
   11 REWIND 42
      IF(TOPT.NE.0)CALL TEMPT(TOPT)
      REWIND 42
      CALL SETUP(A,B)
C***    SOLVE FOR DISPLACEMENTS
      CALL BACSUB(A,B,NEQ,BW,JRAN)
      REWIND 45
      DO 14 J=1,JRAN
      J1=JR1-JRAN+J
      NEQJ = NEQ(J)
   14 WRITE(45) (A(J1,N1),N1=1,NEQJ)
      REWIND 45
      WRITE(41,213)
      DO 15 J=1,JRAN
```

```
      N2= NEQ(J)/2
      READ(45)  (U(N1,J),W(N1,J) ,N1=1,N2)
      I1= IMIN(J)
      I2= IMAX(J)
      J1= JMIN-1+J
      DO 15 I=I1,I2
      I3= I+1-IMIN(J)
      NNN=0
      IF(J.EQ.JRAN.AND.I.EQ.I2)NNN=-1
      WRITE(46) R(I3,J),Z(I3,J),U(I3,J),W(I3,J),NNN
      RPDR=R(I3,J)+U(I3,J)
      ZPDZ=Z(I3,J)+W(I3,J)
   15 WRITE(41,214) I,J1, R(I3,J), Z(I3,J), U(I3,J), W(I3,J), RPDR, ZPDZ
C**** CALCULATION OF ELEMENT STRESSES
      REWIND 44
      JRAN1= JMAX-JMIN
      IC=0
      DO 1501 I=1,14
      SSMAX(I)=-1.E20
 1501 SSMIN(I)=1.E20
      DO 16 J=1,JRAN1
      I1=IMIN(J)
      I2=IMAX(J)-1
      DO 16 I=I1,I2
      I3=I+1-I1
      J3=JMIN-1+J
      IN=I-IMIN(J+1)+1
   16 IF(CODE(I3,J)/1000000.LE.25)CALL STRESS(U,W,R,Z,I3,J,I,J3,CODE,IC,
     1  IN,SSMAX,SSMIN,IJSS)
      XX=1.E20
      NNN=-1
      WRITE(46)I3,JRAN,(XX,I=1,16),NNN
      WRITE(41,222) (IJSS(I,1),IJSS(I,2),SSMIN(I),IJSS(I,3),
     1   IJSS(I,4),SSMAX(I),I=1,14)
  222 FORMAT(1H112X59HMINIMUM AND MAXIMUM VALUES OF STRESS AND STRAIN IN
     1 THE BODY/1H041X7HMINIMUM23X,7HMAXIMUM/1H0,8HQUANTITY25X1HI,4X1HJ
     210X5HVALUE14X1HI,4X1HJ10X5HVALUE/
     317H0RADIAL STRESS       13X2I5,1PE15.4,10X2I5,E15.4/
     417H0HOOP STRESS         13X2I5,E15.4,10X2I5,E15.4/
     517H0AXIAL STRESS        13X2I5,E15.4,10X2I5,E15.4/
     617H0R-Z SHEAR STRESS    13X2I5,E15.4,10X2I5,E15.4/
     717H0MAX STRESS          13X2I5,E15.4,10X2I5,E15.4/
     817H0MIN STRESS          13X2I5,E15.4,10X2I5,E15.4/
     917H0MAX SHEAR STRESS    13X2I5,E15.4,10X2I5,E15.4/
     117H0RADIAL STRAIN       13X2I5,E15.4,10X2I5,E15.4/
     117H0HOOP STRAIN         13X2I5,E15.4,10X2I5,E15.4/
     217H0AXIAL STRAIN        13X2I5,E15.4,10X2I5,E15.4/
     317H0R-Z SHEAR STRAIN    13X2I5,E15.4,10X2I5,E15.4/
     417H0MAX STRAIN          13X2I5,E15.4,10X2I5,E15.4/
     517H0MIN STRAIN          13X2I5,E15.4,10X2I5,E15.4/
     617H0MAX SHEAR STRAIN    13X2I5,E15.4,10X2I5,E15.4)
      READ(40,100) (TITLE(I),I=1,13)
```

```
      2001 CONTINUE
           IF(WORD1.NE.TITLE(13))WRITE(46)TITLE
           IF(WORD2.NE.TITLE(1)) GO TO 1000
           CALL CLEAN
           CALL EXIT
       100 FORMAT(13A6)
       101 FORMAT(8I5)
       102 FORMAT(2I5,2F10.5,I5)
       103 FORMAT(I10,6F10.5/7F10.5)
       104 FORMAT(2I10,4F10.5)
       105 FORMAT(7F10.5/10X4F10.5)
       201 FORMAT(2I10,2F20.4,I10)
       203 FORMAT(1H125X38HM A T E R I A L    P R O P E R T I E S )
       204 FORMAT(10H0MAT. NO.=I2,4X3HE1=1PE11.4,4X5HNU21=E11.4,2X7HALPHA1=
          1  E11.4,5X4HG13=E11.4,5X4HPHI=E11.4/
          2  16X3HE2=E11.4,4X5HNU31=E11.4,2X7HALPHA2=
          2  E11.4,5X4HBFR=E11.4/16X3HE3=E11.4,4X5HNU32=E11.4,2X7HALPHA3=
          3  E11.4,5X4HBFZ= E11.4)
      205  FORMAT(49H1AMG054 FINITE ELEMENT STRESS ANALYSIS OF NOZZLES/
          184H0WRITTEN BY JOHN BRISBANE, ROHM AND HAAS CO., REDSTONE ARSENAL
          2RESEARCH LABORATORIES/1H013A6)
       206 FORMAT( 6H0JMIN=I5/6H JMAX=I5/
          1 18H NO. OF MATERIALS=I5/
          225H NO. OF MATERIAL TABLES = I3)
       207 FORMAT(1H116X34HA P P L I E D    P R E S S U R E S /
          19X1HI9X1HJ13X1HP17X3HTAU11X5HPCODE )
       208 FORMAT(2I10,2F20.5,I10)
       212 FORMAT(I5,52H DATA ERRORS NOTED, PROGRAM PROCEEDS TO NEXT PROBLEM)
       213 FORMAT(1H13X1HI4X1HJ6X12HR-COORDINATE6X12HZ-COORDINATE6X14HR-DISPL
          1ACEMENT6X14HZ-DISPLACEMENT6X11HR + DELTA-R6X11HZ + DELTA-Z /)
       214 FORMAT( 2I5, 2F15.4, 2( 10X, 1PE15.4), 0P2F15.4)
       215 FORMAT( 1H1,35X,49HM A T E R I A L    P R O P E R T Y    T A B L E
          1 S)
       216 FORMAT(1H0/13H MATERIAL NO.I3,10X4HBFR=1PE13.6,10X4HBFZ=E13.6,
          1 10X4HPHI=E13.6/1H0,7X,1HT,7X,2HE1,10X,2HE2,10X,2HE3,8X,4HNU21,
          2 4X,4HNU31,4X,4HNU32,6X,3HG13,8X,6HALPHA1,6X,6HALPHA2,6X,6HALPHA3)
      217  FORMAT(1H F9.0,3E12.4,3F8.5,4E12.4)
      218  FORMAT(35H0THE BODY HAS A UNIFORM TEMPERATURE)
      219  FORMAT(40H0THE TEMPERATURE TABLE WAS INPUT ON TAPE)
      220  FORMAT(41H0THE TEMPERATURE TABLE WAS INPUT ON CARDS)
      221  FORMAT(68H0NODAL POINT TEMPERATURES WERE INPUT ON TAPE AS DETERMIN
          1ED BY AMG065)
           STOP
           END
           SUBROUTINE PROP(E1,E2,E3,NU21,NU31,NU32,G13,PHI,ALPHA1,
          1  ALPHA2,ALPHA3,C,PSI)
           IMPLICIT REAL*8(A-H,O-Z)
           REAL*8 TITLE,WORD1,WORD2,WORD3
           REAL NU21,NU31,NU32
           DIMENSION C(4,4),PSI(4),CD(4,4),D(4,4)
           DATA D(2,2),D(1,2),D(3,2),D(4,2),D(2,1),D(2,3),D(2,4)/1.,6*0./
      C*** FORM THE C MATRIX
```

```
      SIN(X)=DSIN(X)
      COS(X)=DCOS(X)
      CF11 = 1./E2/E3-NU32*NU32/E2/E2
      CF12 = -NU21/E1/E3-NU31*NU32/E1/E2
      CF13 = NU21*NU32/E1/E2+NU31/E1/E2
      CF22 = 1./E1/E3-NU31*NU31/E1/E1
      CF23 = -NU32/E1/E2-NU21*NU31/E1/E1
      CF33 = 1./E1/E2-NU21*NU21/E1/E1
      DET = 1./E1*CF11+NU21/E1*CF12-NU31/E1*CF13
      C(1,1)=CF11/DET
      C(1,2)=-CF12/DET
      C(1,3)=CF13/DET
      C(2,2)=CF22/DET
      C(2,3)=-CF23/DET
      C(3,3)=CF33/DET
      C(4,4)=G13
      C(1,4)=0.
      C(2,4)=0.
      C(3,4)=0.
      C(2,1)=C(1,2)
      C(3,1)=C(1,3)
      C(4,1)=0.
      C(3,2)=C(2,3)
      C(4,2)=0.
      C(4,3)=0.
C***  FORM THE VECTOR PSI
      DO 1 N=1,3
    1 PSI(N)=C(N,1)*ALPHA1+C(N,2)*ALPHA2+C(N,3)*ALPHA3
      PSI(4)=0.
      IF(PHI.EQ.0.) GO TO 8
C***  ROTATE C AND PSI BY THE ANGLE PHI
      PHI=PHI/57.29578
      CP=COS(PHI)
      SP=SIN(PHI)
      D(1,1)=CP*CP
      D(1,3)=SP*SP
      D(1,4)=SP*CP
      D(3,1)=D(1,3)
      D(3,3)=D(1,1)
      D(3,4)=-D(1,4)
      D(4,1)=2.*D(3,4)
      D(4,3)=-D(4,1)
      D(4,4)=D(1,1)-D(1,3)
      DO 2 I=1,4
      CD(I,1)=0.
      DO 2 K=1,4
    2 CD(I,1)=CD(I,1)+D(K,I)*PSI(K)
      DO 3 I=1,4
    3 PSI(I)=CD(I,1)
      DO 5 I=1,4
      DO 5 J=1,4
      CD(I,J)=0.
```

```
      DO 5 K=1,4
    5 CD(I,J)=CD(I,J)+C(I,K)*D(K,J)
      DO 7 I=1,4
      DO 7 J=I,4
      C(I,J)=0.
      DO 6 K=1,4
    6 C(I,J)=C(I,J)+D(K,I)*CD(K,J)
    7 C(J,I)=C(I,J)
    8 RETURN
      END
      SUBROUTINE STIFFQ(I,J,PI,PT,IPT)
C*****    CALCULATION OF STIFFNESS MATRIX OF A QUADRILATERAL COMPOSED
C         OF FOUR TRIANGLES.
C         CENTER NODAL DISPLACEMENTS AND ELEMENT PRESSURE ARE ELIMINATED
C         AND DATA FOR THEIR CALCULATION IS WRITTEN ON TAPE 12
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 TITLE,WORD1,WORD2,WORD3
      INTEGER  ERR,CODE,     TOPT
      COMMON UF,WF,TANF,JMIN,JMAX,ERR,MAX,R,Z,CODE,TITLE,IMAX,IMIN,
     1  CONPR(16,15),DUM1,S,F,NPCARD,TOPT,NP,MN
      DIMENSION UF(100),WF(100),TANF(100),R(30,50),Z(30,50),CODE(30,50),
     1   TITLE(13),IMAX(50),IMIN(50),DUM1( 726),          S(8,8),F(8)
     2  ,IT(6,4),AK(10,10),F2(2),RR(5),C(4,4),ET(4),ZZ(5),F1(10)
      DOUBLE PRECISION  AK,F1,F2,DET
      DATA IT(1,1),IT(2,1),IT(3,1),IT(4,1),IT(5,1),IT(6,1),IT(1,2),
     1IT(2,2),IT(3,2),IT(4,2),IT(5,2),IT(6,2),IT(1,3),IT(2,3),IT(3,3),
     2IT(4,3),IT(5,3),IT(6,3),IT(1,4),IT(2,4),IT(3,4),IT(4,4),IT(5,4),
     3IT(6,4)/1,2,3,4,9,10,3,4,7,8,9,10,7,8,5,6,9,10,5,6,1,2,9,10/
 1000 FORMAT(7H0STIFFQ)
      I1= I+IMIN(J)-1
      IN=I1-IMIN(J+1)+1
      RR(1)=R(I,J)
      RR(2)=R(I+1,J)
      RR(3)=R(IN+1,J+1)
      RR(4)=R(IN,J+1)
      ZZ(1)=Z(I,J)
      ZZ(2)=Z(I+1,J)
      ZZ(3)=Z(IN+1,J+1)
      ZZ(4)=Z(IN,J+1)
      RK= (RR(1)+RR(2)+RR(3)+RR(4))/4.0
      ZK= (ZZ(1)+ZZ(2)+ZZ(3)+ZZ(4))/4.0
      RR(5)=RR(1)
      ZZ(5)=ZZ(1)
      DT=0.
      IF(TOPT.NE.0)READ(42)DT
 1001 FORMAT(8H0STIFFQ1,I10)
      IF(MN.LT.16) GO TO 9
      CALL INTERP(C,ET,BFR,BFZ,DT,MN)
 1002 FORMAT(7H0INTERP)
      ET(1) = ET(1)*DT
      ET(2) = ET(2)*DT
      ET(3) = ET(3)*DT
```

```
      ET(4) = ET(4)*DT
      GO TO 10
    9 ICOUNT=0
      DO 8 II=1,4
      ET(II)=CONPR(II+10,MN)*DT
      DO 8 JJ=II,4
      ICOUNT=ICOUNT+1
      C(II,JJ)=CONPR(ICOUNT,MN)
    8 C(JJ,II)=C(II,JJ)
      BFR=CONPR(15,MN)
      BFZ=CONPR(16,MN)
   10 IPT1=IPT/10
      IPT2=IPT-10*IPT1
      DO 6 N=1,10
      DO 7 M=1,10
    7 AK(N,M)=0.
    6 F1(N)=0.
      DO  1  N=1,4
      P=0.
      TAU=0.
      RI=RR(N)
      ZI=ZZ(N)
      RJ=RR(N+1)
      ZJ=ZZ(N+1)
      IF(IPT1.EQ.N.OR.IPT2.EQ.N) P=PI
      IF(IPT1.EQ.N.OR.IPT2.EQ.N) TAU=PT
 1003 FORMAT(9HOGOSTIFF3)
   13 CALL STIFF3(RI,RJ,RK,ZI,ZJ,ZK,P,TAU,IPT,C,ET,BFR,BFZ)
 1004 FORMAT(10HORESTIFF3 )
      DO 1 M1=1,6
      N1=IT(M1,N)
      F1(N1)=F1(N1)+F (M1)
      DO 1 M2=1,6
      N2=IT(M2,N)
    1 AK(N1,N2)=AK(N1,N2)+S (M1,M2)
      DET=AK(9,9)*AK(10,10)-AK(9,10)**2
 1005 FORMAT(5HODET=,E20.8)
      AK(9,10)=AK(9,9)/DET
      AK(9,9)=AK(10,10)/DET
      AK(10,10)=AK(9,10)
      AK(10,9)=-AK(10,9)/DET
      AK(9,10)=AK(10,9)
      DO 3 N=9,10
      F2(N-8)=0.
      DO  2  M=1,8
      AK(N,M)=0.
      DO 2 N1=9,10
    2 AK(N,M)=AK(N,M)+AK(N,N1)*AK(M,N1)
      DO 3 M=1,2
    3 F2(N-8)=F2(N-8)+AK(N,M+8)*F1(M+8)
      DO  5 N=1,8
      DO 401 M=1,8
```

```
      DET=AK(N,M)
      DO 4 N1=9,10
    4 DET=DET-AK(N,N1)*AK(N1,M)
  401 S(N,M)=DET
      DO 402 M=1,2
  402 F1(N)=F1(N)-AK(N,M+8)*F2(M)
    5 F(N)=F1(N)
      WRITE(44) C,ET,RK,ZK,F2,((AK(I1,I2),I1=9,10),I2=1,8)
      PI=0.
      PT=0.
      IPT=0
 1006 FORMAT(9HORESTIFFQ)
      RETURN
      END
      SUBROUTINE INTERP(C,ET,BFR,BFZ,TEM ,MN)
C***  INTERPERTING IN MATERIAL PROPERTY TABLE
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 TITLE,WORD1,WORD2,WORD3
      DIMENSION C(4,4),ET(4),TABLE(12,10),X(10)
      COMMON/VARPRO/XTABLE(12,10,5)
      EQUIVALENCE (X,E1),(X(2),E2),(X(3),E3),(X(4),XNU21),(X(5),XNU31),(
     1  X(6),XNU32),(X(7),G13),(X(8),ALPHA1),(X(9),ALPHA2),(X(10),ALPHA3
     2  )
      DO 8 II=1,12
      DO 8 JJ=1,10
    8 TABLE(II,JJ)=XTABLE(II,JJ,MN-15)
    1 NTEMP=TABLE(12,1)+0.5
      TEMP=TEM +TABLE(12,5)
      IF(TEMP.LE.TABLE(1,1))GO TO 5
      IF(TEMP.GE.TABLE(1,NTEMP))GO TO 6
      DO 2 N=2,NTEMP
      N1=N
      IF(TEMP.LE.TABLE(1,N).AND.TEMP.GE.TABLE(1,N-1))GO TO 3
    2 CONTINUE
    3 RATIO=(TEMP-TABLE(1,N1-1))/(TABLE(1,N1)-TABLE(1,N1-1))
    7 CONTINUE
      DO 4 II=2,11
    4 X(II-1)=TABLE(II,N1-1)+RATIO*(TABLE(II,N1)-TABLE(II,N1-1))
      PHI=TABLE(12,6)
      BFR=TABLE(12,2)
      BFZ=TABLE(12,3)
      CALL PROP(E1,E2,E3,XNU21,XNU31,XNU32,G13,PHI,ALPHA1,ALPHA2,ALPHA3,
     1  C,ET)
      RETURN
    5 N1=2
      RATIO=0.
      GO TO 7
    6 N1=NTEMP
      RATIO=1.0
      GO TO 7
      END
      SUBROUTINE MESH
```

```
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 R1,Z1,W1,W2,W3
      REAL*4 POLAR, SAME
      REAL*8 TITLE,WORD1,WORD2,WORD3
      REAL LINE
      INTEGER  CT,CODE,TYPE,ERR,PCODE,BW
      DIMENSION R1(30),Z1(30)
      DIMENSION R(30,50),Z(30,50),CODE(30,50),TITLE(13),UF(100),WF(100)
     1  ,TANF(100),IMAX(50),IMIN(50),BC(100,3),CONPR(16,15),IP(200)
     2  ,JP(200),P(200),TAU(200),PCODE(200),NEQ(50),           S(8,8)
     3  ,F(8),RR(5),ZZ(5)
      COMMON UF,WF,TANF,JMIN,JMAX,ERR,MAX,R,Z,CODE,TITLE,IMAX,IMIN,
     1        CONPR,IP,JP,P,TAU,PCODE,BW,NEQ,JRAN,        S,F,NPCARD
     2    ,ITOPT,NP,MN
      EQUIVALENCE  (BC(1),UF)
      DATA POLAR,LINE,SAME/1HP,1HL,1HS/
      DATA WORD1/6HH ONLY/
C****   READ NODAL POINT DATA
      ABS(X)=DABS(X)
      SIN(X)=DSIN(X)
      COS(X)=DCOS(X)
      WRITE(41,204)
      ICOUNT=0
      IC=1
      IF(ITOPT.NE.4.AND. ITOPT .NE. 5) GO TO 1
      JRAN=JMAX+1-JMIN
      READ(42) (IMAX(J),IMIN(J),NEQ(J),J=1,JRAN),BW
      BW=2*BW
      DO 1000 J=1,JRAN
      IRAN=NEQ(J)
      NEQ(J) = 2*NEQ(J)
      READ(42)(R1(I),Z1(I),CODE(I,J),I=1,IRAN)
      DO 1100 I=1,IRAN
      R(I,J)=R1(I)
 1100 Z(I,J)=Z1(I)
 1000 CONTINUE
 1001 READ(40,104) W2,I,J,TYPE,I1,I2,I3,I4,BC1,BC2,BC3,BC4
      IF(I.EQ.0) GO TO 9
      IF(J.LE.JMAX ) GO TO 1002
      ERR=ERR+1
      WRITE(41,200) I,J
 1002 CT=I4+10*I3+100*I2+1000*I1
      IF(I1+I2+I3+I4.NE.0) WRITE(41,203) I,J,I1,BC1,I2,BC2,I4,BC4
      IF(ABS(BC1)+ABS(BC2)+ABS(BC3)+ABS(BC4).EQ.0.) GO TO 1003
      UF(IC)=BC1
      WF(IC)=BC2
      TANF(IC)=BC4
      CT=CT+IC*10000
      IC=IC+1
      IF(IC.LE.100) GO TO 1003
      ERR=ERR+1
      WRITE(41,201)
```

```
      GO TO 19
 1003 CT=CT+1000000*TYPE
      J1=J-JMIN+1
      I8=I-IMIN(J1)+1
      CODE(I8,J1)=(CODE(I8,J1)/100000000)*100000000+CT
      IF(W2.NE.LINE) GO TO 1001
      READ(40,104) W2,II,JJ
      NSTEPS=MAX0(IABS(II-I),IABS(JJ-J))
      ISTEP=(II-I)/NSTEPS
      JSTEP=(JJ-J)/NSTEPS
      DO 1004 N=1,NSTEPS
      I=I+ISTEP
      J=J+JSTEP
      IF(I1+I2+I3+I4.NE.0) WRITE(41,203) I,J,I1,BC1,I2,BC2,I4,BC4
      J1=J+1-JMIN
      I8=I-IMIN(J1)+1
1004  CODE(I8,J1)=(CODE(I8,J1)/100000000)*100000000+CT
      GO TO 1001
  104 FORMAT(1X,A1,I3,2I5,4I1,21X,4F10.0)
    1 READ(40,100) W1,W2,I,J,TYPE,I1,I2,I3,I4,RT,ZT,BC1,BC2,BC3,BC4
      IF(I.EQ.0) GO TO 5
      KK=0
  101 IF(J.LE.JMAX) GO TO 2
      ERR= ERR+1
      WRITE(41,200)I,J
    2 CT=I4+10*I3+100*I2+1000*I1
      IF(I1+I2+I3+I4.NE.0)WRITE(41,203)I,J,I1,BC1,I2,BC2,        I4,BC4
      IF(ABS(BC1)+ABS(BC2)+ABS(BC3)+ABS(BC4).EQ.0.) GO TO 3
      UF(IC) = BC1
      WF(IC) = BC2
      TANF(IC)=BC4
      CT=CT+IC*10000
      IC = IC+1
      IF(IC.LE.100)  GO TO 3
      ERR = ERR+1
      WRITE(41,201)
      GO TO 19
    3 CT = CT+1000000*TYPE
      IF(ABS(RT)+ABS(ZT).NE.0.) CT=CT+100000000
      J1=J-JMIN+1
      IMIN(J1) = MIN0(IMIN(J1),I)
      IMAX(J1) = MAX0(IMAX(J1),I)
      IF(W1.NE.POLAR) GO TO 4
      RAD = RT
      ANGLE=ZT/57.2957795
      RT= RAD*COS(ANGLE)
      ZT= RAD*SIN(ANGLE)
    4 WRITE(45) I,J1,RT,ZT,CT
      ICOUNT=ICOUNT+1
      IF(KK.EQ.1) GO TO 404
      IF(W2.NE.LINE) GO TO 401
C****  CALCULATION OF POINTS ON A STRAIGHT LINE
```

```
      READ(40,103) W1,W3,II,JJ,RT1,ZT1
      IF(W1.NE.POLAR) GO TO 402
      RAD = RT1
      ANGLE = ZT1/57.2957795
      RT1= RAD*COS(ANGLE)
      ZT1=RAD*SIN(ANGLE)
      W1 = 0.
  402 NSTEPS=MAX0(IABS(II-I),IABS(JJ-J))
      ISTEP=(II-I)/NSTEPS
      JSTEP=(JJ-J)/NSTEPS
      DD=NSTEPS
      DR = (RT1-RT)/DD
      DZ= (ZT1-ZT)/DD
      DO 404 N=1,NSTEPS
      IJK = N
      I=I+ISTEP
      J=J+JSTEP
      J1=J-JMIN+1
      IMAX(J1)=MAX0(IMAX(J1),I)
      IMIN(J1)=MIN0(IMIN(J1),I)
      RT=RT+DR
      ZT=ZT+DZ
      IF(W3.EQ.SAME) GO TO 403
      READ(40,102) I,J,TYPE,I1,I2,I3,I4,BC1,BC2,BC3,BC4
      KK=1
      GO TO 101
  403 CT=CT
      IF(J.EQ.JMAX) CT=MOD(CT,1000000)+130000000
      WRITE(45) I,J1,RT,ZT,CT
      ICOUNT =ICOUNT+1
      IF(I1+I2+I3+I4.NE.0) WRITE(41,203) I,J,I1,BC1,I2,BC2,I4,BC4
  404 CONTINUE
  401 GO TO 1
C***   STORE COORDINATES AND CODE IN  I,J ARRAYS
    5 REWIND 45
   51 READ(45) I,J1,RT,ZT,CT
      ICOUNT=ICOUNT-1
      I1=I-IMIN(J1)+1
      R(I1,J1)=RT
      Z(I1,J1)=ZT
      IF (I .EQ. IMAX(J1)) CT = MOD(CT,1000000) + 130000000
      CODE(I1,J1)=CT
      IF(ICOUNT.GT.0)GO TO 51
      JRAN = JMAX-JMIN+1
C***    DETERMINATION OF BAND WIDTH
      MAXBAN = 80
      J2= JRAN-1
      BW= 0
      DO 52 J=1,J2
      NEQ(J)= 2*(IMAX(J)+1-IMIN(J))
      NBAND= 2*(IMAX(J)+3-IMIN(J+1))
      IF(NBAND.LE.MAXBAN) GO TO 52
```

```
      WRITE(41,207) J1,MAXBAN
   52 BW= MAX0(BW,NBAND)
      NEQ(JRAN)=2*(IMAX(JRAN)+1-IMIN(JRAN))
      DO 6 J=1,JRAN
      IF(IMAX(J)-IMIN(J).LE.MAX-1) GO TO 6
      ERR= ERR+1
      J1 = J+JMIN-1
      WRITE(41,202) J1
    6 CONTINUE
      IF(ERR.NE.0) GO TO 19
C****   CALCULATE COORDINATES OF INTERIOR GRID POINTS
      IF(JRAN.LE.2) GO TO 9
      J2= JRAN-1
      DO 8 N=1,500
      IJK = N
      RESID= 0.
      DO 7 J=2,J2
      IRAN = IMAX(J)-IMIN(J)
      IN = IMIN(J)-IMIN(J+1)+1
      IM = IMIN(J)-IMIN(J-1)+1
      DO 7 I=2,IRAN
      IN = IN+1
      IM = IM+1
      IF(CODE(I,J).GE.100000000) GO TO 7
      DR = (R(IN,J+1)+R(IM,J-1)+R(I+1,J)+R(I-1,J))/4.-R(I,J)
      DZ= (Z(IM,J-1)+Z(IN,J+1)+Z(I+1,J)+Z(I-1,J))/4.-Z(I,J)
      R(I,J) = R(I,J)+1.8*DR
      Z(I,J) = Z(I,J)+1.8*DZ
      RESID = RESID + ABS(DR)+ABS(DZ)
    7 CONTINUE
      IF(N.EQ.1) RES1= RESID
      IF(RESID/RES1.LT.1.E-5) GO TO 9
    8 CONTINUE
C****   OUTPUT OF NODAL POINT COORDINATES
        WRITE(43,799)
799     FORMAT(6X,°I°,3X,°J°,12X,°MAT NO.°,12X,°AREA°,/)
    9 WRITE(41,205)N
      DO 16 J=1,JRAN
      I1=IMIN(J)-1
      J1= JMIN-1+J
      IRAN = IMAX(J)-I1
      DO 16 I=1,IRAN
      I1= I1+1
C***   FIX RADIAL DISPLACEMENT IF R EQUALS ZERO
      IF(R(I,J).EQ.0.)CODE(I,J)=(CODE(I,J)/10000)*10000+1000+MOD(CODE(
     1 I,J),1000)
C***   SET TYPE EQUAL 30 IF I EQUAL IRAN OR J EQUALS JMAX
      IF(I.EQ.IRAN.OR.J.EQ.JRAN)CODE(I,J)=CODE(I,J)/100000000*100000000
     U  +30000000 +MOD(CODE(I,J),1000000)
      WRITE(41,206) I1,J1,R(I,J),Z(I,J),CODE(I,J)
        IF(CODE(I,J).EQ.0) GO TO 225
      TYPE=(CODE(I,J)-100000000 )/1000000
```

```
      GO TO 226
  225 TYPE=0
  226 WRITE(46) I,J,TYPE
   16 CODE(I,J) = MOD(CODE(I,J),100000000)
      I=0
      J=0
      TYPE=0
      WRITE(46) I,J,TYPE
C***  CALCULATION OF AREAS OF TRIANGLES
      DO 18 J=1,JRAN
      I1= IMIN(J)
      I2= IMAX(J)
      J1= JMIN-1+J
      DO 18 I= I1,I2
      I3= I-I1+1
      IF(CODE(I3,J)/1000000.GT.25)GO TO 18
      IN= I-IMIN(J+1)+1
      RR(1)= R(I3,J)
      RR(2)= R(I3+1,J)
      RR(3)= R(IN+1,J+1)
      RR(4)= R(IN,J+1)
      RR(5)= RR(1)
      ZZ(1)= Z(I3,J)
      ZZ(2)= Z(I3+1,J)
      ZZ(3)= Z(IN+1,J+1)
      ZZ(4)= Z(IN,J+1)
      ZZ(5)= ZZ(1)
      RK= (RR(1)+RR(2)+RR(3)+RR(4))/4.
      ZK= (ZZ(1)+ZZ(2)+ZZ(3)+ZZ(4))/4.
        AREAT=0.
        ARM=0.
      DO 17 N=1,4
      AREA = (RR(N+1)-RR(N))*(ZK-ZZ(N))-(RK-RR(N))*(ZZ(N+1)-ZZ(N))
        RA=(RR(N)+RR(N+1)+RK)/3.
        ARM=RA*AREA+ARM
      IF(AREA.GT.0.) GO TO 17
      ERR= ERR+1
      WRITE(41,220) N,I,J1
17      AREAT=AREAT+AREA
        NTT=(CODE(I+1-IMIN(J),J)/1000000)
        IF(NTT.NE.18) GO TO 18
        VOL=3.1415926536*ARM
        WRITE(43,999) I,J1,VOL,NTT
999     FORMAT(2X,2I5,5X,°VOL=°,E10.4,I15)
   18 CONTINUE
      IF(TITLE(13).NE.WORD1)WRITE(46) IMAX,IMIN,JRAN
   19 RETURN
  100 FORMAT(2A1,I3,2I5,4I1,1X,6F10.5)
  102 FORMAT(3I5,4I1,21X,4F10.5)
  103 FORMAT(2A1,I3,I5,10X,2F10.5)
  200 FORMAT(26H0J EXCEEDS JMAX ON CARD I=I5,4H  J=I5)
  201 FORMAT(53H0MORE THAN 99 NODES HAVE NON ZERO BOUNDARY CONDITIONS)
```

```
  202 FORMAT(36H0MORE THAN 30 NODAL POINTS ON ROW J=I5 )
  203 FORMAT(2I5,3(I8,1PE17.7))
  204 FORMAT(25H0BOUNDARY CONDITION ARRAY/10H0 NODAL PT10X6HRADIAL19X5HA
     1XIAL                    15X7HSLIDING /1H 3X1HI4X1HJ5X4HCODE7X5HVALUE9X
     24HCODE7X5HVALUE9X4HCODE7X5HVALUE)
  205 FORMAT(30H1COORDINATES CALCULATED AFTER I3,11H ITERATIONS /
     14X1HI4X1HJ10X1HR14X1HZ14X4HCODE /)
  206 FORMAT(2I5,2F15.4,I15)
  207 FORMAT(21H BAND WIDTH OF ROW J=I3, 8HEXCEEDS I5)
  220 FORMAT(49H ZERO OR NEGATIVE AREA IN TRIANGULAR ELEMENT NO. I1,
     126H OF QUADRILATERAL ELEMENT I2,2H, I2 )
      END
      SUBROUTINE  TRIAN(A,B,NEQJ,NEQJ1,BW)
C****    THIS SUBROUTINE TRIANGLIZES A BLOCK OF BANDED EQUATIONS
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 TITLE,WORD1,WORD2,WORD3
      INTEGER  BW,BW1
      DIMENSION A(120,80),B(120)
      DOUBLE PRECISION RATIO
      N3= NEQJ+NEQJ1
      DO 1 N=1,NEQJ
      N1= N+1
      N2= MIN0(BW+N-1,N3)
      IF(A(N,1).EQ.0.) GO TO 2
      DO 1 I=N1,N2
      I1= I+1-N
      RATIO= A(N,I1)/A(N,1)
      IF(RATIO.EQ.0.)GO TO 1
      B(I)= B(I)-RATIO*B(N)
      J1= BW-I1+1
      DO 3 J=1,J1
      I2= I-N+J
    3 A(I,J)= A(I,J)-A(N,I2)*RATIO
    1 CONTINUE
      RETURN
    2 WRITE(41,201)
  201 FORMAT(49H ZERO TERM ON MAJOR DIAGONAL EXECUTION TERMINATED)
      STOP
      END
      SUBROUTINE BACSUB(A,B,NEQ,BW,JRAN)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 TITLE,WORD1,WORD2,WORD3
      INTEGER BW
      DIMENSION A(120,80),B(120),NEQ(50),DISP(120)
      DOUBLE PRECISION DP
      DATA JR1/121/
      DO 5 J=1,JRAN
      J1= JRAN+1-J
      NEQJ= NEQ(J1)
      NEQJ1= NEQ(J1+1)
      IF(J1.EQ.JRAN) NEQJ1= 0
      N1= NEQJ+NEQJ1
```

```
      BACKSPACE 45
      READ(45) ((A(I1,I2),I2=1,BW),B(I1),I1=1,NEQJ)
      BACKSPACE 45
      DO 2 N=1,NEQJ
      N2= NEQJ+1-N
      DP= B(N2)
      DO 1 K=2,BW
      K1= N2-1+K
      IF(K1.GT.N1) GO TO 2
    1 DP= DP-A(N2,K)*DISP(K1)
    2 DISP(N2) = DP/A(N2,1)
      JR = JR1-J
      DO 3 K=1,NEQJ
    3 A(JR,K)= DISP(K)
      IF(J.EQ.JRAN)GO TO 5
      DO 4 K=1,NEQJ
      K1= NEQ(J1-1)+K
    4 DISP(K1) = A(JR,K)
    5 CONTINUE
      RETURN
      END
      SUBROUTINE  INVRT(A,ACT,DIM)
C       INVERSION OF SYMMETRIC MATRIX
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 TITLE,WORD1,WORD2,WORD3
      INTEGER ACT,DIM
      DIMENSION A(DIM,DIM),LOC(61)
      DOUBLE PRECISION DP
      ABS(X)=DABS(X)
      DO 1 N=1,ACT
    1 LOC(N)=N
      DO 6 N1=1,ACT
      M=0
      PIVOT=0.
      DO 2 N2=N1,ACT
      NN=LOC(N2)
      IF (ABS(A(NN,NN)).LE.ABS(PIVOT)) GO TO 2
      M=N2
      PIVOT=A(NN,NN)
    2 CONTINUE
      IF (M.EQ.0) GO TO 8
      N=LOC(M)
      LOC(M)=LOC(N1)
      LOC(N1)=N
      A(N,N)=-1.
      DO 3 J=1,ACT
    3 A(N,J)=A(N,J)/PIVOT
      DO 5 I1=1,ACT
      I=LOC(I1)
      DP=A(I,N)
      IF (N.EQ.I.OR.A(I,N).EQ.0.) GO TO 5
      DO 4 J1=I1,ACT
```

```
      J=LOC(J1)
      IF (N.EQ.J) GO TO 4
      A(I,J)=A(I,J)-       A(N,J)*DP
      A(J,I)=A(I,J)
    4 CONTINUE
    5 CONTINUE
      DO 6 I=1,ACT
    6 A(I,N)=A(N,I)
      DO 7 I=1,ACT
      DO 7 J=1,ACT
    7 A(I,J)=-A(I,J)
      RETURN
    8 WRITE(41,9)
    9 FORMAT (42H0MATRIX IS SINGULAR - EXECUTION TERMINATED  )
      CALL EXIT
      RETURN
      END
      SUBROUTINE CLEAN
      RETURN
      END
      SUBROUTINE TEMPT(TOPT)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 RI,ZI,TIMP,TXME,TXEMP
      REAL*8 TITLE,WORD1,WORD2,WORD3 ,ISTUFF
      INTEGER TOPT,CODE,ERR
      LOGICAL L1,L2
      COMMON ISTUFF,JMIN,JMAX,ERR,MAX,R,Z,CODE,TITLE,IMAX,IMIN
      DIMENSION RI(2000),ZI(2000),TIMP(2000)
      DIMENSION TEMP(2000),RT(2000),ZT(2000),D(5),RR(5),ZZ(5),T(5)
     1  ,ISTUFF(300),R(30,50),Z(30,50),CODE(30,50),TITLE(13)
     2  ,IMAX(50),IMIN(50),DT2(30)
     3  ,TTEMP(30,50),IPRINT(30)
      COMMON/TEM/A(120,80)
      EQUIVALENCE (A(3001),RT),(A(5001),ZT),(A(7001),TEMP,TTEMP)
      DIMENSION TXEMP(30,50)
      SIN(X)=DSIN(X)
      COS(X)=DCOS(X)
      ABS(X)=DABS(X)
      ATAN(X)=DATAN(X)
      READ(40,100) T0
      WRITE(41,201)T0
      ITOPT=TOPT+1
      GO TO(20,1,3,4,41),ITOPT
    1 READ(40,100) T1
      DT= T1-T0
      WRITE(41,202)DT
      JRAN1= JMAX-JMIN
      DO 2 J=1,JRAN1
      I1= IMIN(J)
      I2=IMAX(J)-1
      DO 2 I=I1,I2
      I3= I+1-I1
```

```
      IF(CODE(I3,J)/1000000.GT.25) GO TO 2
      WRITE(42) DT
    2 CONTINUE
      RETURN
    3 READ(40,101) NOTEMP,Z0,D4
      READ(42) (RI(N),ZI(N),TIMP(N),N=1,NOTEMP)
      DO 999 N=1,NOTEMP
      RT(N)=RI(N)
      ZT(N)=ZI(N)
  999 TEMP(N)=TIMP(N)
      GO TO 5
    4 READ(40,101) NOTEMP,Z0,D4
      READ(40,102) (RT(N),ZT(N),TEMP(N),N=1,NOTEMP)
      GO TO 5
   41 READ(40,101)NDIST
      DO 42 N1=1,NDIST
      READ(42) TXME,TXEMP
      TIME=TXME
      DO 442L3=1,30
      DO 442 L4=1,50
  442 TTEMP(L3,L4)= TXEMP(L3,L4)
   42 CONTINUE
      GO TO 602
    5 DO 6 N=1,NOTEMP
    6 ZT(N)= ZT(N)-Z0
  602 JRAN1= JMAX-JMIN
      D4=D4*D4
      IF(TOPT.NE.4)WRITE(41,203)
      IF(TOPT.EQ.4)WRITE(41,205)TIME
      DO 16 J=1,JRAN1
      I1= IMIN(J)
      I2= IMAX(J)-1
      J1=JMIN-1+J
      DO 1601 I=I1,I2
      I3= I+1-I1
      IF(CODE(I3,J)/1000000.GT.25) GO TO 1601
      IN= I-IMIN(J+1)+1
      RK= (R(I3,J)+R(I3+1,J)+R(IN,J+1)+R(IN+1,J+1))/4.
      ZK= (Z(I3,J)+Z(I3+1,J)+Z(IN,J+1)+Z(IN+1,J+1))/4.
      IF(ITOPT.NE.5)GO TO 1602
      RR(1)=R(I3,J)
      RR(2)=R(I3+1,J)
      RR(3)=R(IN+1,J+1)
      RR(4)=R(IN,J+1)
      ZZ(1)=Z(I3,J)
      ZZ(2)=Z(I3+1,J)
      ZZ(3)=Z(IN+1,J+1)
      ZZ(4)=Z(IN,J+1)
      T(1)=TTEMP(I3,J)
      T(2)=TTEMP(I3+1,J)
      T(3)=TTEMP(IN+1,J+1)
      T(4)=TTEMP(IN,J+1)
```

```
      GO TO 1201
 1602 I4=I3
      I5 = IN
      IF (I .LT. (I1 + I2)/2) GO TO 601
      I4 = I4 + 1
      I5 = I5 + 1
  601 THETA=1.570795
      IF(ABS(R(I5,J+1)-R(I4,J)).GT..001)THETA=ATAN((Z(I5,J+1)-Z(I4,J))/(
     1 R(I5,J+1)-R(I4,J)))
      D(1)=D4
      D(2)=D4
      D(3)=D4
      D(4)=D4
      DO 12 N=1,NOTEMP
      AA=RT(N)-RK
      BB=ZT(N)-ZK
      DD=AA*AA+BB*BB
      IF(DD.GE.D4)GO TO 12
      L1=.TRUE.
      L2=.TRUE.
      IF(AA*SIN(THETA)-BB*COS(THETA).LT.0.)L1=.FALSE.
      IF(AA*COS(THETA)+BB*SIN(THETA).LT.0.)L2=.FALSE.
      DO 11 L=1,4
      GO TO(7,8,9,10),L
    7 IF(.NOT.L1.OR..NOT.L2) GO TO 11
      GO TO 10
    8 IF(L1.OR..NOT.L2) GO TO 11
      GO TO 10
    9 IF(L1.OR.L2) GO TO 11
      GO TO 10
 10   IF(DD.GE.D(L))GO TO 12
      D(L)=DD
      T(L)=TEMP(N)
      RR(L)= RT(N)
      ZZ(L)= ZT(N)
      GO TO 12
   11 CONTINUE
   12 CONTINUE
 1201 T(5)=T(1)
      CC= 0.
      DT1= 0.
      RR(5)= RR(1)
      ZZ(5)= ZZ(1)
      D(5)=D(1)
      DO 17 N=1,4
      IF(ITOPT.EQ.5)GO TO 1701
      IF(D(N).GE.D4.OR.D(N+1).GE.D4) GO TO 17
 1701 AJ= RR(N+1)-RR(N)
      BJ= ZZ(N+1)-ZZ(N)
      AK= RK-RR(N)
      BK= ZK-ZZ(N)
      AREA=AJ*BK-AK*BJ
```

```
      WRITE(41,599) AJ,BK,AK,BJ,AREA
599   FORMAT(2X,°AJ,BK,AK,BJ,AREA=°,5E13.6)
      IF(AREA.EQ.0.) GO TO 17
      C= ZZ(N+1)-ZK
      DX= RK-RR(N+1)

      COMM= (RR(N)+RR(N+1)+RK)/6./AREA
      DT1= COMM*(BK*BJ+AJ*AK)*T(N+1)+COMM*(C*BJ-DX*AJ)*T(N)+DT1

      CC=CC+COMM*(BJ*BJ+AJ*AJ)
   17 CONTINUE

      DT1= DT1/CC-T0
      WRITE(42) DT1
      IPRINT(I3)=I
 1601 DT2(I3)=DT1

      IRAN=I2+1-I1

   16 WRITE(41,204) J1,(IPRINT(I),DT2(I),I=1,IRAN)
   20 RETURN

100     FORMAT(F10.5)
  101 FORMAT(I10, 2F10.5)
  102 FORMAT(2F10.5,E15.6)
 201  FORMAT(1H1,20X,48HT E M P E R A T U R E   D I S T R I B U T I O N
     1   //10X23HREFRENCE TEMPERATURE IS F10.3)
 202  FORMAT(65HOTHE TEMPERATURE DIFFERENCE IS UNIFORM THROUGHOUT THE BO
     1DY AND IS  F10.3 )
 203  FORMAT(64HOTHE TEMPERATURE DIFFERENCES WERE DETERMINED FROM AN INP
     1UT TABLE)
  204 FORMAT(7HOROW J=I3/(3H I=I3,2X3HDT=F8.2,

     1  4X2HI=I3,2X3HDT=F8.2,
     1  4X2HI=I3,2X3HDT=F8.2,
     1  4X2HI=I3,2X3HDT=F8.2,
     1  4X2HI=I3,2X3HDT=F8.2))
 205  FORMAT(72HOTHE TEMPERATURE DIFFERENCES WERE DETERMINED FROM THE DI
     1STRIBUTION AT T=,1PE12.5,19H AS FOUND BY AMG065)
      END
      SUBROUTINE STRESS(U,W,R,Z,I,J,I3,J3,CODE,IC,IN,SSMAX,SSMIN,IJSS)
C*****     CALCULATION OF STRESSES AND STRAINS IN A QUADRILATERAL
```

```
C         DISPLACEMENT FIELD IS FITTED BY A FOUR TERM PARABALOID
C         TO THE FOUR CORNER AND CENTER POINT DISPLACEMENTS BY
C         LEAST SQUARES
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 TITLE,WORD1,WORD2,WORD3
      DOUBLE PRECISION  F2,S28
      INTEGER CODE
      LOGICAL EL1,EL2
      EQUIVALENCE (EP,EPR),(EP(2),EPT),(EP(3),EPZ),(EP(4),GAMRZ),(SIG,
     1SIGR),(SIG(2),SIGT),(SIG(3),SIGZ),(SIG(4),TAURZ)
      EQUIVALENCE (SS,SIG),(SS(5),SIGMAX),(SS(6),SIGMIN),(SS(7),TAUMAX),
     1  (SS(8),EP),(SS(12),EPMAX),(SS(13),EPMIN),(SS(14),GAMMAX)
        DIMENSION  PHI(4,5),PTP(4,4),PU(4),PW(4),F2(2),S28(2,8)
     1  ,U1(5),W1(5),S4(4,4),ALF(4),BET(4),U(30,50),W(30,50)
     2  ,C(4,4),EP(4),ET(4),SIG(4),CODE(30,50),R(30,50),Z(30,50)
     3  ,SSMAX(14),SSMIN(14),SS(14),IJSS(14,4)
      DATA PHI(1,1),PHI(1,2),PHI(1,3),PHI(1,4),PHI(1,5) /5*1./,
     1PHI(2,1),PHI(3,1),PHI(4,1)/3*0.0 /
      COS(X)=DCOS(X)
      SIN(X)=DSIN(X)
      ABS(X)=DABS(X)
      SQRT(X)=DSQRT(X)
      ATAN(X)=DATAN(X)
      SIGN(X,Y)=DSIGN(X,Y)
      READ(44) C,ET,RR,ZZ,F2,S28
      U1(2)=U(I,J)
      U1(3)=U(I+1,J)
      U1(4)=U(IN,J+1)
      U1(5)=U(IN+1,J+1)
      W1(2)=W(I,J)
      W1(3)=W(I+1,J)
      W1(4)=W(IN,J+1)
      W1(5)=W(IN+1,J+1)
      PHI(2,2)=R(I,J)-RR
      PHI(2,3)=R(I+1,J)-RR
      PHI(2,4)=R(IN,J+1)-RR
      PHI(2,5)=R(IN+1,J+1)-RR
      PHI(3,2)=Z(I,J)-ZZ
      PHI(3,3)=Z(I+1,J)-ZZ
      PHI(3,4)=Z(IN,J+1)-ZZ
      PHI(3,5)=Z(IN+1,J+1)-ZZ
      DO 1  M=2,5
    1 PHI(4,M)=PHI(2,M)**2 + PHI(3,M)**2
      DO 2 M=1,2
      DO 2  MM=2,5
    2 F2(M)=F2(M)-S28(M,2*MM-3)*U1(MM)-S28(M,2*MM-2)*W1(MM)
      U1(1)=F2(1)
      W1(1)=F2(2)
      DO 4  N=1,4
      DO 3  M=1,4
      S4(N,M)=0.
      DO 3  MM=1,5
```

```
    3 S4(N,M)=S4(N,M)+PHI(N,MM)*PHI(M,MM)
      PU(N)=0.
      PW(N)=0.
      DO 4 M=1,5
      PU(N)=PU(N)+PHI(N,M)*U1(M)
    4 PW(N)=PW(N)+PHI(N,M)*W1(M)
      CALL INVRT(S4,4,4)
      DO 5 N=1,4
      ALF(N)=0.
      BET(N)=0.
      DO 5  M=1,4
      ALF(N)=ALF(N)+S4(N,M)*PU(M)
    5 BET(N)=BET(N)+S4(N,M)*PW(M)
      EPR=ALF(2)
      EPT = U1(1)/RR
      EPZ=BET(3)
      GAMRZ=ALF(3)+BET(2)
      ANGLE=.5*ATAN(    GAMRZ/( EPR- EPZ))*57.2958
      IF (EPR .LT. EPZ) ANGLE = ANGLE + SIGN(90.0D0,GAMRZ)
      TEM=(EPR +EPZ )/2.
      TEM1 =SQRT((( EPR- EPZ)/2.)**2+GAMRZ**2/4.)
      EPMAX=TEM+TEM1
      EPMIN=TEM-TEM1
      GAMMAX=2.*TEM1
      DO 8 N=1,4
      SIG(N)=-ET(N)
      DO 8  M= 1,4
    8 SIG(N)= SIG(N)+C(N,M)*EP(M)
      TEM = (SIGR+SIGZ)/2.
      TEM1 = SQRT(((SIGR-SIGZ)/2.)**2+TAURZ**2)
      SIGMAX= TEM+TEM1
      SIGMIN= TEM-TEM1
      TAUMAX = TEM1
      IF  (MOD(IC,19)) 7,6,7
    6 WRITE(41,100)
    7 WRITE(41,101)I3,J3,RR,ZZ,SIGR,SIGT,SIGZ,TAURZ,SIGMAX,SIGMIN,
     1TAUMAX,ANGLE,EPR,EPT,EPZ,GAMRZ,EPMAX,EPMIN,GAMMAX,CODE(I,J)
        IF(CODE(I,J).NE.18000000) GO TO 661
        WRITE(47,499) I3,J3,SIGR,SIGT,SIGMAX,SIGMIN,CODE(I,J)
  499   FORMAT(1H0I3,I4,1P4E13.4,I15)
  661 IC=IC+1
      DO 11 K=1,14
      IF(SS(K).GT.SSMIN(K))GO TO 10
      SSMIN(K)=SS(K)
      IJSS(K,1)=I3
      IJSS(K,2)=J3
   10 IF(SS(K).LT.SSMAX(K))GO TO 11
      SSMAX(K)=SS(K)
      IJSS(K,3)=I3
      IJSS(K,4)=J3
   11 CONTINUE
      EL1=.FALSE.
```

```
      EL2=.FALSE.
      IF (CODE(IN,J+1)/1000000 .LT. 26) EL1 = .TRUE.
      IF (CODE(I+1,J )/1000000 .LT. 26) EL2 = .TRUE.
      IF (.NOT.EL1.AND..NOT.EL2) N=1
      IF (.NOT.EL1.AND.EL2) N = 2
      IF (EL1.AND..NOT.EL2) N=3
      IF (EL1.AND.EL2) N = 4
      WRITE(46) I,J,RR,ZZ,SIGR,SIGZ,SIGT,TAURZ,SIGMAX,SIGMIN,TAUMAX
     1    ,EPR,EPZ,EPT,GAMRZ,EPMAX,EPMIN,GAMMAX,N
  100 FORMAT( 9H1   I    J/
     1                 5X 11HCOORDINATES37X33HS T R E S S E S  /  S T R
     1A I N S  /8H   ANGLE7X1HR8X1HZ4X9HRADIAL   R2X10HHOOP THETA5X8HAXIA
     2L  Z3X10HSHEAR   R-Z6X7HMAXIMUM6X7HMINIMUM4X9HMAX SHEAR )
  101 FORMAT(1H0I3,I4,0PF8.3,F9.3,1P7E13.4/1H 0PF7.2,17X1P7E13.4,I15)
  102 FORMAT (1H 1P11E11.3)
      RETURN
      END
      SUBROUTINE PRESBC(ICF,JCF,H,TE,SIDE,LC,ERR)
C***    THIS SUBROUTINE READS THE PRESSURE BOUNDARY CONDITION DATA
C       DATA MAY BE READ IN FOR AN ELEMENT SIDE AT A TIME OR FOR A LINE
C       WHICH HAS I OR J AS A CONSTANT. THE SUBROUTINE MAY BE CHANGED
C       TO FIT PARTICULAR NEEDS.
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 TITLE,WORD1,WORD2,WORD3
      INTEGER  SIDE,      SIDET,      ERR
C
      DIMENSION           ICF(200),       JCF(200),       H(200),
     1 TE(200),           SIDE(200)
C
      LC=0
    1 READ(40,100) I1,J1,I2,J2,HT,TET,SIDET
      IF(I1.EQ.0)RETURN
      LC=LC+1
      IF(LC.GT.200) GO TO 5
      ICF(LC)=I1
      JCF(LC)=J1
      H(LC)=HT
      TE(LC)=TET
      SIDE(LC)=SIDET
      IF(I2.EQ.0) GO TO 1
      NSTEPS=MAX0(IABS(I2-I1),IABS(J2-J1))
      ISTEP=(I2-I1)/NSTEPS
      JSTEP=(J2-J1)/NSTEPS
      DO 3 N=1,NSTEPS
      LC=LC+1
      IF(LC.GT.200) GO TO 5
      ICF(LC)=ICF(LC-1)+ISTEP
      JCF(LC)=JCF(LC-1)+JSTEP
      H(LC)=HT
      TE(LC)=TET
    3 SIDE(LC)=SIDET
      GO TO 1
```

```
    5 ERR=ERR+1
      WRITE(41,209)
      RETURN
  100 FORMAT(4I5,2F10.5,I5)
  209 FORMAT(70HONUMBER OF ELEMENT SIDES WITH PRESSURE BOUNDARY CONDITIO
     1NS EXCEEDS 200)
      END
      SUBROUTINE SETUP(A,B)
C***    ASSEMBLE STIFFNESS MATRIX OF STRUCTURE IN THE FORM OF A BAND
C       EQUATIONS ARE MODIFIED FOR BOUNDARY CONDITIONS AND
C       TRIANGLIZED BEFORE BEING WRITTEN ON TAPE
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 TITLE,WORD1,WORD2,WORD3
      INTEGER BW,BCCODE,SLCODE,UCODE,WCODE,CODE,BW1,ERR,PCODE
      COMMON BC,        JMIN,JMAX,ERR,MAX,R,Z,CODE,TITLE,IMAX,IMIN,
     1       CONPR,IP,JP,P,TAU,PCODE,BW,NEQ,JRAN,       S,F,NPCARD
     2  ,ITOPT,NP,MN
      EQUIVALENCE(IFIX(1),UCODE),(IFIX(2),WCODE),(BC(1),UF),(BC(101),WF)
     1   ,(BC(201),TANF)
      DIMENSION R(30,50),Z(30,50),CODE(30,50),TITLE(13),UF(100),WF(100)
     1  ,TANF(100),IMAX(50),IMIN(50),BC(100,3),CONPR(16,15),IP(200)
     2  ,JP(200),P(200),TAU(200),PCODE(200),NEQ(50),           S(8,8)
     3  ,A(120,80),B(120),IFIX(2),F(8)
      NP=1
      REWIND 45
      IPT=0
      DO 1 I=1,120
      B(I)= 0.
      DO 1 J=1,80
    1 A(I,J)= 0.
      DO 20 J=1,JRAN
      IRAN= NEQ(J)/2
      IF(J.EQ.JRAN) GO TO 6
      DO 501 I=1,IRAN
      MN1= CODE(I,J)/1000000
      IF(MN1.GT.0.AND.MN1.LT.26)MN=MN1
      IF(MN1.GT.25) GO TO 501
C*****     CHECK FOR PRESSURE ON THE ELEMENT
      I1= IMIN(J)+I-1
      J1= JMIN+J-1
      IF(NPCARD.LT.NP)GO TO 1002
      IS=IP(NP)
      JS=JP(NP)
      IF(IS.NE.I1.OR.JS.NE.J1)GO TO 1002
      PI=P(NP)
      PT=TAU(NP)
      IPT=PCODE(NP)
      NP=NP+1
 1002 CALL STIFFQ(I,J,PI,PT,IPT)
C***    ADD ELEMENT STIFFNESS TO TOTAL STIFFNESS
      N1= 2*(I-1)
      N2= 2*(I+IMIN(J)-1-IMIN(J+1))+NEQ(J)
```

```
      DO 5 N=1,4
        N3 = N2-N1+1-N
      NN1= N1+N
      NN2= N2+N
      B(NN1)= B(NN1)+F(N)
      B(NN2)= B(NN2)+F(N+4)
      LL= 0
      DO 4 L= N,4
      LL= LL+1
      A(NN1,LL)= A(NN1,LL)+S(N,L)
    4 A(NN2,LL)= A(NN2,LL)+S(N+4,L+4)
      DO 5 L=1,4
      L1= N3+L
    5 A(NN1,L1)= A(NN1,L1)+S(N,L+4)
  501 CONTINUE
C***    MODIFY THE BLOCK OF EQUATIONS FOR FIXITY AND APPLIED LOADS
    6 DO 11 I=1,IRAN
      BCCODE= MOD(CODE(I,J),1000000)
      IF(BCCODE.EQ.0) GO TO 11
      SLCODE= MOD(BCCODE,10)
      WCODE= MOD(BCCODE,1000)/100
      UCODE= MOD(BCCODE,10000)/1000
      NCODE= MOD(BCCODE,1000000)/10000
      IF(NCODE.EQ.0) NCODE=100
      IU= 2*(I-1)+1
      IW= IU+1
      IF(UCODE.EQ.2) B(IU)= B(IU)+UF(NCODE)/6.2831853
      IF(WCODE.EQ.2) B(IW)= B(IW)+WF(NCODE)/6.2831853
      IF(SLCODE.EQ.0) GO TO 8
      ALF= TANF(NCODE)
      B(IU)= B(IU)+ALF*B(IW)
      B(IW)= 0.
      A(IU,1)= A(IU,1)+ALF*(ALF*(A(IW,1)+1.)+2.*A(IU,2))
      A(IU,2)= -ALF
      A(IW,1)= 1.
      BW1= BW-1
      DO 7 N=2,BW1
      A(IU,N+1)= A(IU,N+1)+ALF*A(IW,N)
      A(IW,N)= 0.
      II= IU+1-N
      IF(II.LT.1) GO TO 7
      A(II,N)= A(II,N)+ALF*A(II,N+1)
      A(II,N+1)= 0.
    7 CONTINUE
      A(IW,BW)= 0.
    8 NMAX= NEQ(J)+NEQ(J+1)
      IF (J .EQ. JRAN) NMAX = NEQ(J)
      DO 10 N=1,2
      IR=IU+N-1
      IF(IFIX(N).NE.1) GO TO 10
      DO 9 N1=2,BW
      II= IR+1-N1
```

```
      IJ= IR+N1-1
      IF(II.GT.0)B(II)= B(II)-A(II,N1)*BC(NCODE,N)
      IF(IJ.LE.NMAX)B(IJ)= B(IJ)-A(IR,N1)*BC(NCODE,N)
      IF(II.GT.0) A(II,N1)= 0.
    9 A(IR,N1)= 0.
      A(IR,1)= 1.
      B(IR)= BC(NCODE,N)
   10 CONTINUE
   11 CONTINUE
      IF(J.EQ.JRAN) GO TO 17
      IRAN1= IMAX(J+1)+1-IMIN(J+1)
      DO 16 I=1,IRAN1
      BCCODE= MOD(CODE(I,J+1),1000000)
      IF(BCCODE.EQ.0) GO TO 16
      SLCODE= MOD(BCCODE,10)
      WCODE= MOD(BCCODE,1000)/100
      UCODE= MOD(BCCODE,10000)/1000
      NCODE= MOD(BCCODE,1000000)/10000
      IF(NCODE.EQ.0) NCODE= 100
      IU= NEQ(J)+2*(I-1)+1
      IW= IU+1
      IF(SLCODE.EQ.0) GO TO 13
      ALF= TANF(NCODE)
      BW1=BW-1
      DO 12 N=2,BW1
      II= IU+1-N
      IF(II.GT.NEQ(J).OR.II.LT.1) GO TO 12
      A(II,N)= A(II,N)+ALF*A(II,N+1)
      A(II,N+1)= 0.
   12 CONTINUE
   13 DO 15 N=1,2
      IF(IFIX(N).NE.1) GO TO 15
      IR=IU+N-1
      DO 14 N1=2,BW
      II=IR+1-N1
      IF(II.LT.1.OR.II.GT.NEQ(J)) GO TO 14
      B(II)= B(II)-A(II,N1)*BC(NCODE,N)
      A(II,N1)= 0.
   14 CONTINUE
   15 CONTINUE
   16 CONTINUE
C***    TRIANGLIZE THE J BLOCK OF COEFFICIENTS
   17 NEQJ= NEQ(J)
      NEQJ1= NEQ(J+1)
      IF(J.EQ.JRAN)NEQJ1=0
      N4 = NEQJ+NEQJ1
      CALL TRIAN(A,B,NEQJ,NEQJ1,BW)
      WRITE(45)((A(N1,N2),N2=1,BW),B(N1),N1=1,NEQJ)
      IF(J.EQ.JRAN)GO TO 20
      DO 18 N1=1,NEQJ1
      N3= NEQJ+N1
      B(N1)= B(N3)
```

```
      DO 18 N2=1,BW
   18 A(N1,N2)= A(N3,N2)
      NEQJ=NEQJ1+1
      J1= JMIN+J-1
      ERR= ERR+1
      DO 19 N1= NEQJ,N4
      B(N1)= 0.
      DO 19 N2=1,BW
   19 A(N1,N2)= 0.
   20 CONTINUE
      RETURN
      END
      SUBROUTINE  STIFF3(RI,RJ,RK,ZI,ZJ,ZK,P,TAU,IPT,C,ET,BFR,BFZ)
C     CALCULATION OF STIFFNESS OF AN ANISOTROPIC TRIANGULAR ELEMENT
C     ELEMENT WITH ANISOTROPIC PROPERTIES AND LINEAR DISPLACEMENTS
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 TITLE,WORD1,WORD2,WORD3
      COMMON  ISTUFF,S,F,NPCARD,ITOPT,NP,MN
      DIMENSIONISTUFF(10162),S(8,8),F(8),A(4,6),CA(4,6),C(4,4),
     1  ET(4),CODE(30,50)
      DATA A(1,1),A(1,2),A(1,3),A(1,4),A(1,5),A(1,6),A(2,1),A(2,2),A(2,3
     1),A(2,4),A(2,5),A(2,6),A(3,1),A(3,2),A(3,3),A(3,4),A(3,5),A(3,6),
     2A(4,1),A(4,2),A(4,3),A(4,4),A(4,5),A(4,6)/24*0.0/
      ABS(X)=DABS(X)
      DO 10 I = 1,7
      F(I) = 0.
      DO 10 J = 1,7
   10 S(I,J) = 0.
      DEL=(RJ-RI)*(ZK-ZI)-(RK-RI)*(ZJ-ZI)
    1 A(1,1)=(ZJ-ZK)/DEL
      A(1,3)=(ZK-ZI)/DEL
      A(1,5)=-A(1,3)-A(1,1)
      A(2,1)=1./3./RI
      A(2,3)=1./3./RJ
      A(2,5)=1./3./RK
      A(3,2)=(RK-RJ)/DEL
      A(3,4)=(RI-RK)/DEL
      A(3,6)=-A(3,2)-A(3,4)
      A(4,1)=A(3,2)
      A(4,2)=A(1,1)
      A(4,3)=A(3,4)
      A(4,4)=A(1,3)
      A(4,5)=A(3,6)
      A(4,6)=A(1,5)
      DO  2  I=1,4
      DO  2  J=1,6
      CA(I,J)=0.
      DO  2  K=1,4
    2 CA(I,J)=CA(I,J)+C(I,K)*A(K,J)
      DO  4  I=1,6
      DO  3  J=I,6
      DO  3  K=1,4
```

```
    3 S(I,J)=S(I,J)+A(K,I)*CA(K,J)
      DO  4  J=1,4
    4 F(I)=F(I)+A(J,I)*ET(J)
      VOL = DEL*(RI+RJ+RK)/6.
      DO 42 I = 1,6
      F(I) = F(I)*VOL
      DO 42 J = I,6
      S(I,J) = S(I,J)*VOL
   42 S(J,I) = S(I,J)
      IF (ABS(P)+ABS(TAU)+ABS(BFR)+ABS(BFZ).EQ.0.) GO TO 5
      AJ= RJ-RI
      AK= RK-RI
      BJ= ZJ-ZI
      BK= ZK-ZI
      IF(ABS(BFR)+ABS(BFZ).EQ.0.)GO TO 41
      X1=(RI+RJ+RK)/6.
      X2=(RI*(RI+RJ+RK)+RJ*(RJ+RK)+RK*RK)/12.
      X3=((RI*RI+RJ*RJ+RK*RK)*(RI+RJ+RK)+RI*RJ*RK)/20.
      X4=(BK*(RI+2.*RK+RJ)+BJ*(RI+2.*RJ+RK))/24.
      X5=(BK*(RK*(2.*(RI+RJ)+3.*RK)+RI*(RI+RJ)+RJ*RJ)+
     1      BJ*(RJ*(2.*(RI+RK)+3.*RJ)+RI*(RI+RK)+RK*RK))/60.
   41 F(1)= F(1)+(TAU*AJ-P*BJ)*(RI/2.+AJ/6.)+BFR*((RJ*BK-RK*BJ)*X2+
     1          (BJ-BK)*X3+(AK-AJ)*X5)
      F(2)= F(2)+(TAU*BJ+P*AJ)*(RI/2.+AJ/6.)+BFZ*((RJ*BK-RK*BJ)*X1+
     1          (BJ-BK)*X2+(AK-AJ)*X4)
      F(3)= F(3)+(TAU*AJ-P*BJ)*(RI/2.+AJ/3.)+BFR*(-RI*BK*X2+BK*X3-AK*X5)
      F(4)= F(4)+(TAU*BJ+P*AJ)*(RI/2.+AJ/3.)+BFZ*(-RI*BK*X1+BK*X2-AK*X4)
      F(5)= F(5)+BFR*(RI*BJ*X2-BJ*X3+AJ*X5)
      F(6)= F(6)+BFZ*(RI*BJ*X1-BJ*X2+AJ*X4)
    5 CONTINUE
      RETURN
      END
```